PLXMon 2000
Software Development Utility
User's Manual

PLXMon 2000
Software Development Utility
User's Manual

Version 3.1

May 2000

Website: http://www.plxtech.com
Email: apps@plxtech.com
Phone: 408 774-9060
800 759-3735
Fax: 408 774-2169

© 2000, PLX Technology, Inc. All rights reserved.

PLX Technology, Inc. retains the right to make changes to this product at any time, without notice. Products may have minor variations to this publication. PLX assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of PLX products.

This document contains proprietary and confidential information of PLX Technology Inc. (PLX). The contents of this document may not be copied nor duplicated in any form, in whole or in part, without prior written consent from PLX Technology, Inc.

PLX provides the information and data included in this document for your benefit, but it is not possible for us to entirely verify and test all of this information in all circumstances, particularly information relating to non-PLX manufactured products. PLX makes no warranties or representations relating to the quality, content or adequacy of this information. Every effort has been made to ensure the accuracy of this manual, however, PLX assumes no responsibility for any errors or omissions in this document. PLX shall not be liable for any errors or for incidental or consequential damages in connection with the furnishing, performance, or use of this manual or the examples herein. PLX assumes no responsibility for any damage or loss resulting from the use of this manual; for any loss or claims by third parties which may arise through the use of this SDK; and for any damage or loss caused by deletion of data as a result of malfunction or repair. The information in this document is subject to change without notice.

PLX Technology and the PLX logo are registered trademarks of PLX Technology, Inc.

Other brands and names are the property of their respective owners.

Document number: PLXMON2000-SDK-MAN-P1-3.1

PLX SOFTWARE LICENSE AGREEMENT

THIS PLX SOFTWARE IS LICENSED TO YOU UNDER SPECIFIC TERMS AND CONDITIONS. CAREFULLY READ THE TERMS AND CONDITIONS PRIOR TO USING THIS SOFTWARE. OPENING THIS SOFTWARE PACKAGE OR INITIAL USE OF THIS SOFTWARE INDICATES YOUR ACCEPTANCE OF THE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD RETURN THE ENTIRE SOFTWARE PACKAGE TO PLX.

LICENSE Copyright © 2000 PLX Technology, Inc.

This PLX Software License agreement is a legal agreement between you and PLX Technology, Inc. for the PLX Software, which is provided on the enclosed PLX CD-ROM. PLX Technology owns this PLX Software. The PLX Software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties, and is licensed, not sold. If you are a rightful possessor of the PLX Software, PLX grants you a license to use the PLX Software as part of or in conjunction with a PLX chip on a **per project basis**. PLX grants this permission provided that the above copyright notice appears in all copies and derivatives of the PLX Software. Use of any supplied runtime object modules or derivatives from the included source code in any product without a PLX Technology, Inc. chip is strictly prohibited. You obtain no rights other than those granted to you under this license. You may copy the PLX Software for backup or archival purposes. You are not authorized to use, merge, copy, display, adapt, modify, execute, distribute or transfer, reverse assemble, reverse compile, decode, or translate the PLX Software except to the extent permitted by law.

PLX Software License Agreement

GENERAL

If you do not agree to the terms and conditions of this PLX Software License Agreement, do not install or use the PLX Software and promptly return the entire unused PLX Software to PLX Technology, Inc. You may terminate your PLX Software license at any time. PLX Technology may terminate your PLX Software license if you fail to comply with the terms and conditions of this License Agreement. In either event, you must destroy all your copies of this PLX Software. Any attempt to sub-license, rent, lease, assign or to transfer the PLX Software except as expressly provided by this license, is hereby rendered null and void**.**

WARRANTY

PLX Technology, Inc. provides this PLX Software AS IS, WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, AND ANY WARRANTY OF MERCHANTIBILITY OR FITNESS FOR A PARTICULAR PURPOSE. PLX makes no guarantee or representations regarding the use of, or the results based on the use of the software and documentation in terms of correctness, or otherwise; and that you rely on the software, documentation, and results solely at your own risk. In no event shall PLX be liable for any loss of use, loss of business, loss of profits, incidental, special or, consequential damages of any kind. In no event shall PLX's total liability exceed the sum paid to PLX for the product licensed here under.

PLX Copyright Message Guidelines

The following copyright message along with the following text must appear in all software products generated and distributed, which use the PLX API libraries:

"Copyright © 2000 PLX Technology, Inc."

Requirements:

- Arial font
- Font size 12 (minimum)
- Bold type
- Must appear as shown above in the first section or the so called "Introduction Section" of all manuals
- Must also appear as shown above in the beginning of source code as a comment

TABLE OF CONTENTS

1. Introduction 1-1

1.1 ABOUT THIS MANUAL 1-1
1.2 CONVENTIONS AND SUPPORT 1-1
1.3 INSTALLATION 1-2
1.4 PLXMON 2000 FEATURE LIST 1-2
1.5 CUSTOMER SUPPORT INFORMATION 1-3

2. A Brief Tour of PLXMon 2000 2-1

2.1 STARTING PLXMON 2000 AND CONFIGURATION 2-1
2.1.1 Default Vendor and Device IDs for PLX Reference Design Boards 2-3
2.2 DISPLAYING REGISTERS 2-3
2.3 POPULAR FEATURES 2-3
2.3.1 EEPROM Edit Utility 2-4
2.3.2 Downloading To The IOP 2-4
2.3.3 DMA Transfers 2-5
2.4 BOARD WITH NO LOCAL CPU 2-6
2.4.1 Using PCI 9054RDK-LITE Board or Your Custom Board with No CPU 2-6

3. PLXMon 2000 - Reference 3-1

3.1 ACCESS MODE 3-1
3.1.1 PCI Mode 3-1
3.1.2 IOP (Serial) Mode 3-1
3.2 APPLICATION HOT LINKS 3-2
3.3 DEVICE CONFIGURATION 3-2
3.4 DOWNLOADING IOP APPLICATIONS 3-3
3.5 FONT CONFIGURATION 3-4
3.6 THE INTERFACE 3-4
3.6.1 The PLXMon 2000 Toolbar 3-5
3.6.2 Status Bar 3-5
3.6.3 Command Line Interface (CLI) 3-5
3.6.3.1 Displaying Memory Via Memory Cycles, (dl, dw, db) 3-5
3.6.3.2 Displaying Memory Via I/O Cycles, (il, iw, ib) 3-5
3.6.3.3 Writing Memory Via Memory Cycles, (el, ew, eb) 3-5
3.6.3.4 Writing Memory Via I/O Cycles, (ol, ow, ob) 3-6
3.6.3.5 The Pci Command 3-6

© PLX Technology, Inc. All rights reserved.

3.6.3.6 The Quit Command 3-6
3.6.3.7 The Reg Command 3-6
3.6.3.8 The Repeat Command (r) 3-6
3.6.3.9 User Variables (vars) 3-7
3.6.3.10 The Ver Command 3-7
3.7 PRINT, PRINT PREVIEW, AND PRINT SETUP 3-7
3.8 REGISTER ACCESS/REGISTER SETS 3-7
3.9 THE RESET BUTTON 3-7
3.10 SELECTING DEVICES 3-8
3.11 SERIAL CONFIGURATION 3-8
3.12 SERIAL EEPROM ACCESS 3-9
3.13 MEMORY ACCESS 3-9

4. The IOP 480 Register Set 4-1

4.1 THE REGISTER GROUP DIALOG BOXES 4-1
4.1.1 PCI Configuration Register Group Dialog Box 4-1
4.1.2 Local Configuration Register Group Dialog Box 4-2
4.1.3 The Runtime Register Group Dialog Box 4-2
4.1.4 The DMA Register Group Dialog Box 4-3
4.1.5 The Messaging Queue Register Group Dialog Box 4-4
4.1.6 IOP 480 Memory Controller register Group Dialog Box 4-5
4.1.7 IOP 480 CPU Registers Group Dialog Box 4-6
4.1.8 IOP 480 EEPROM Values 4-7

5. The PCI 9054 Register Set 5-1

5.1 THE REGISTER GROUP DIALOG BOXES 5-1
5.1.1 PCI Configuration Register Group Dialog Box 5-1
5.1.2 Local Configuration Register Group Dialog Box 5-3
5.1.3 The Runtime Register Group Dialog Box 5-7
5.1.4 The DMA Register Group Dialog Box 5-8
5.1.5 The Messaging FIFO Register Group Dialog Box 5-10

6. The PCI 9080 Register Set 6-1

6.1 THE REGISTER GROUP DIALOG BOXES 6-1
6.1.1 PCI Configuration Register Group Dialog Box 6-1
6.1.2 Local Configuration Register Group Dialog Box 6-2
6.1.3 The Runtime Register Group Dialog Box 6-6
6.1.4 The DMA Register Group Dialog Box 6-7

© PLX Technology, Inc. All rights reserved.

6.1.5 The Messaging FIFO Register Group Dialog Box 6-9

7. The PCI 9030 Register Set 7-1

7.1 THE REGISTER GROUP DIALOG BOXES 7-1

7.1.1 PCI Configuration Register Group Dialog Box 7-1

7.1.2 Local Configuration Register Group Dialog Box 7-2

7.1.3 Chip Select Register Group Dialog Box 7-3

7.1.4 Runtime Register Group Dialog Box 7-3

Appendix A. Troubleshooting A-1

Appendix B. Glossary Of Terms B-1

© PLX Technology, Inc. All rights reserved.

LIST OF FIGURES

FIGURE 2-1. THE PLXMON 2000 INTERFACE 2-1
FIGURE 2-2. SELECTING THE PROPERTIES 2-1
FIGURE 2-3. DEVICE CONFIGURATION 2-2
FIGURE 2-4. EEPROM EDIT UTILITY 2-3
FIGURE 2-5. DOWNLOAD TO IOP 2-4
FIGURE 2-6. PCI 9080 AND 9054 DMA CHANNEL 0 REGISTERS 2-6
FIGURE 2-7. IOP 480 DMA CHANNEL 0 REGISTERS 2-6
FIGURE 2-8. PLXMON 2000 STARTUP SCREEN 2-7
FIGURE 2-9. PCI 9054RDK-LITE BOARD PROPERTIES 2-7
FIGURE 2-10. PROPERTIES FOR BOARD WITH NO CPU 2-8
FIGURE 3-1. PCI VS. SERIAL DATA FLOW 3-1
FIGURE 3-2. APPLICATION HOT LINKS DIALOG BOX 3-2
FIGURE 3-3. DEVICE CONFIGURATION DIALOG BOX 3-2
FIGURE 3-4. FILE DOWNLOAD DIALOG BOX 3-3
FIGURE 3-5. FONT SELECT DIALOG BOX 3-4
FIGURE 3-6. THE PLXMON 2000 INTERFACE 3-4
FIGURE 3-7. COMMAND LINE INTERFACE COMMANDS 3-5
FIGURE 3-8. INTERACTIVE MODE 3-6
FIGURE 3-9. USER VARIABLES AND THEIR DEFINITIONS 3-7
FIGURE 3-10. DEVICE SELECT DIALOG BOX 3-8
FIGURE 3-11. SERIAL COMMUNICATIONS PROPERTIES DIALOG BOX 3-8
FIGURE 3-12. SERIAL EEPROM ACCESS SCREEN 3-9
FIGURE 3-13. MEMORY DISPLAY SCREEN 3-9
FIGURE 3-14. MEMORY FILL OPTIONS DIALOG BOX 3-10
FIGURE 3-15. USER DEFINED DATA PATTERN 3-10
FIGURE 3-16. MEMORY FILLED WITH USER DEFINED DATA PATTERN 3-11
FIGURE 4-1 PCI CONFIGURATION REGISTER GROUP DIALOG BOX 4-1
FIGURE 4-2. LOCAL CONFIGURATION REGISTERS DIALOG BOX 4-2
FIGURE 4-3. RUNTIME REGISTER GROUP DIALOG BOX 4-2
FIGURE 4-4. INTERRUPT ENABLE /STATUS REGISTER DIALOG BOX 4-3
FIGURE 4-5. DMA REGISTER GROUP DIALOG BOX 4-3
FIGURE 4-6. MESSAGING QUEUE REGISTER GROUP 4-4
FIGURE 4-7. IOP 480 MEMORY CONTROLLER REGISTERS 4-5
FIGURE 4-8. IOP 480 CPU REGISTERS GROUP 4-6
FIGURE 4-9. IOP 480 EEPROM VALUES 4-7
FIGURE 5-1. PCI CONFIGURATION REGISTERS DIALOG BOX FOR PCI 9054 5-1
FIGURE 5-2. POWER MANAGEMENT CAPABILITIES DIALOG BOX 5-2
FIGURE 5-3. POWER MANAGEMENT CSR DIALOG BOX 5-2
FIGURE 5-4. HOT SWAP CSR DIALOG BOX 5-2
FIGURE 5-5. LOCAL CONFIGURATION REGISTERS DIALOG BOX 5-3
FIGURE 5-6. MODE/ARBITRATION DIALOG BOX 5-3
FIGURE 5-7. ENDIAN DESCRIPTOR DIALOG BOX 5-4
FIGURE 5-8. LOCAL MISCELLANEOUS CONTROL REGISTER DIALOG BOX 5-4
FIGURE 5-9. REGION 0/ROM DESCRIPTOR DIALOG BOX 5-5
FIGURE 5-10. DIRECT MASTER REMAP DIALOG BOX 5-5
FIGURE 5-11. DIRECT MASTER CONFIG. DIALOG BOX 5-6
FIGURE 5-12. REGION 1 DESCRIPTOR DIALOG BOX 5-6
FIGURE 5-13. RUN TIME REGISTERS DIALOG BOX 5-7
FIGURE 5-14. INTERRUPT CONTROL AND STATUS DIALOG BOX 5-7
FIGURE 5-15. EEPROM PCI USER IO DIALOG BOX 5-8
FIGURE 5-16. LOCAL DMA REGISTERS DIALOG BOX 5-8
FIGURE 5-17. DMA MODE DIALOG BOX 5-9

© PLX Technology, Inc. All rights reserved.

FIGURE 5-18. DMA DESCRIPTOR POINTER DIALOG BOX 5-9
FIGURE 5-19. DMA THRESHOLDS DIALOG BOX 5-9
FIGURE 5-20. MESSAGING UNIT REGISTERS DIALOG BOX 5-10
FIGURE 5-21. STATUS/CONTROL REGISTER DIALOG BOX 5-10
FIGURE 6-1. PCI CONFIGURATION REGISTERS DIALOG BOX 6-1
FIGURE 6-2. LOCAL CONFIGURATION REGISTERS DIALOG BOX 6-2
FIGURE 6-3. DIRECT MASTER MODE / ARBITRATION DIALOG BOX 6-3
FIGURE 6-4. ENDIAN DESCRIPTOR DIALOG BOX 6-3
FIGURE 6-5. LOCAL SPACE 0/EMP ROM DIALOG BOX 6-4
FIGURE 6-6. DIRECT MASTER PCI REMAP DIALOG BOX 6-4
FIGURE 6-7. DIRECT MASTER CONFIGURE DIALOG BOX 6-5
FIGURE 6-8. REGION 1 DESCRIPTOR DIALOG BOX 6-5
FIGURE 6-9. RUNTIME REGISTERS DIALOG BOX 6-6
FIGURE 6-10. INTERRUPT CONTROL AND STATUS DIALOG BOX 6-6
FIGURE 6-11. EEPROM, PCI, USER IO DETAILS DIALOG BOX 6-7
FIGURE 6-12. DMA REGISTERS DIALOG BOX 6-7
FIGURE 6-13. DMA MODE DIALOG BOX 6-8
FIGURE 6-14. DESCRIPTOR POINTER DIALOG BOX 6-8
FIGURE 6-15. DMA CHANNELS THRESHOLD DIALOG BOX 6-8
FIGURE 6-16. MESSAGING UNIT REGISTERS DIALOG BOX 6-9
FIGURE 6-17. FIFO STATUS/CONTROL REGISTER DIALOG BOX 6-9
FIGURE 7-1. PCI CONFIGURATION REGISTERS DIALOG BOX 7-1
FIGURE 7-2. LOCAL CONFIGURATION REGISTERS DIALOG BOX 7-2
FIGURE 7-3. LOCAL MEMORY BUS REGION DESCRIPTOR REGISTERS DIALOG BOX 7-2
FIGURE 7-4. CHIP SELECT REGISTERS DIALOG BOX 7-3
FIGURE 7-5. RUNTIME REGISTERS DIALOG BOX 7-3
FIGURE 7-6. INTERRUPT ENABLE/STATUS DIALOG BOX 7-4
FIGURE 7-7. PCI TARGET RESPONSE 7-5
FIGURE 7-8. GENERAL PURPOSE I/O CONTROL DIALOG BOX 7-6

LIST OF TABLES

TABLE 2-1. RDK BOARDS SUPPORTED BY PLXMON 2000 2-3

© PLX Technology, Inc. All rights reserved.

1. Introduction

The PLXMon 2000 is a powerful Windows-based GUI debug utility that allows easy configuration, viewing, and modification of various registers on PLX's PCI devices. It also allows the user to perform interactive block-mode, scatter-gather, and shuttle mode DMA operations. PLXMon 2000 incorporates a built-in downloader application for downloading application specific code to your target hardware and supports both FLASH and EEPROM programming. PLXMon 2000 works with the Windows 98, Win 2000, and Windows NT 4.0 operating systems. It is designed to operate with PLX's Reference Design Boards and customer specific hardware that incorporates either the IOP 480, PCI 9030, PCI 9054, or PCI 9080 devices. It includes an EEPROM Edit Utility to allow programming of blank EEPROMs.

The dialog box windows contain detailed information about each register and if appropriate about the individual bits within. Combining this information with the programming algorithms included for accessing the various IOP components and the ability to communicate over both PCI bus and Serial port, PLXMon 2000 provides the tools needed to access all PLX RDKs and your prototype or production board, properly.

PLXMon 2000 is automatically installed on your system when you install Host SDK v3.1. It is compatible with PLX's IOP 480, PCI 9030, PCI 9054 and PCI 9080 devices.

1.1 About This Manual

This manual is divided into seven chapters. Chapter 1 is the Introduction; Chapter 2 shows you how to set up PLXMon 2000 quickly, and how some of the more common functions are used. Chapter 3 gives a self-guided tour of PLXMon 2000. In this area every feature found in PLXMon 2000 is outlined. Chapters 4, 5, 6, and 7 describe the PLXMon 2000 GUI screens for the IOP 480, PCI 9054, PCI 9080, and PCI 9030 devices, respectively. Appendix A is the troubleshooting section and Appendix B is the glossary of terms.

1.2 Conventions And Support

References to Windows NT assume Windows NT 4.0 or higher and will be shown as WinNT. Similarly, references to Windows98 or Windows2000 will be shown as Win98 or Win2000.

The PCI Host SDK contains software for a Windows host to access the PLX Chip across the PCI bus. The PCI Pro SDK also contains software for a local CPU, or IOP (I/O Processor) to access the chip via the local bus on a peripheral card. For further details see the documentation for the Pro SDK.

All references to IOP (I/O Processor) throughout this manual refer to the embedded hardware and all references to IOP software refer to the embedded software.

- All values used in the manual are hexadecimal numbers, with the exception for memory sizes (used in Local Configuration Register screen). The prefix '0x' has been omitted from all hexadecimal numbers and is not required when entering values for PLXMon 2000 fields.

It is important to note that PLXMon 2000 can only operate with a PLX device. This version has been designed to work with the following PLX hardware platforms:

- IOP 480RDK
- PCI 9030RDK-LITE
- CompactPCI 9030RDK-LITE
- PCI 9054RDK-LITE
- PCI 9054RDK-860

© PLX Technology, Inc. All rights reserved.

- CompactPCI 9054RDK-860
- PCI 9080RDK-401B
- PCI 9080RDK-860
- Any customer board that uses the IOP 480, PCI 9030, PCI 9080 or PCI 9054 chips.

PLXMon 2000 has been tested to ensure compatibility with Windows NT, Windows98 and Windows2000.

Note: *PLXMon 2000 is designed to run best with a high-resolution display, such as 1024x768 with 256 colors or better. Users choosing to run the software at lower resolutions will find it visually inconvenient.*

1.3 Installation

In most cases, the PLXMon 2000 program is installed on your system when you install Host SDK v3.1. You can launch this program by clicking on the "P-PLXMon 2000" icon in the Host SDK folder in the Start Menu.

1.4 PLXMon 2000 Feature List

PLXMon 2000 provides the following features:

- Graphical User Interface (GUI) screens that are based on PLX device registers
- Compatible with the IOP 480, PCI 9030, PCI 9054 and PCI 9080 chips
- EEPROM Edit Utility to program a blank EEPROM
- Split Screen Interface, allowing command line input while receiving serial data.
- Serial communications with an IOP's debug port. This feature is compatible with PLX's Back-End Monitor protocol
- A built-in downloader providing support for the following image standards: Motorola S-Record, IBM-401B Image Files, COFF, and Binary. This feature supports downloading to RAM and FLASH devices through the PCI bus and the serial port.
- PLX EEPROM Configuration screens to modify the contents of NM93CS46, NM93CS56, and NM93CS66 EEPROMs. These configuration screens allow you to load and save values from and to a file.
- Memory display providing easy access to IOP memory spaces or allocated PCI memory buffer.
- Customizable Hot-Links. This feature allows users to launch Win32 compatible programs such as testing and sample programs

© PLX Technology, Inc. All rights reserved.

1.5 Customer Support Information

Prior to contacting PLX customer support, please ensure that you are situated close to the computer that has the Host SDK installed and have the following information:

1. Model number of the PLX PCI RDK (if any)
2. PLX Host SDK version (if any)
3. Host Operating System and version
4. PLXMon 2000 version
5. Description of your intended design:
    - PLX chip used
    - Microprocessor (if any)
    - Local Operating System and version (if any)
    - I/O devices (if any)
6. Description of your problem
7. Steps to recreate the problem

You may contact PLX customer support at:

Address: PLX Technology, Inc.
Attn. Technical Support
390 Potrero Avenue
Sunnyvale, CA 94086

Phone: 408-774-9060
Fax: 408-774-2169
Web: http://www.plxtech.com

You may send email to one of the following addresses:

west-apps@plxtech.com
mid-apps@plxtech.com
east-apps@plxtech.com
euro-apps@plxtech.com
asia-apps@plxtech.com

© PLX Technology, Inc. All rights reserved.

2. A Brief Tour of PLXMon 2000

This section will give a 10-minute tour of PLXMon 2000. During this tour you will become familiar with how to setup the program. The more common features found in PLXMon 2000 will also be discussed.

2.1 Starting and Configuring PLXMon 2000

PLXMon 2000 can be started by:

- Clicking on the icon in the Host SDK folder in the Start Menu.

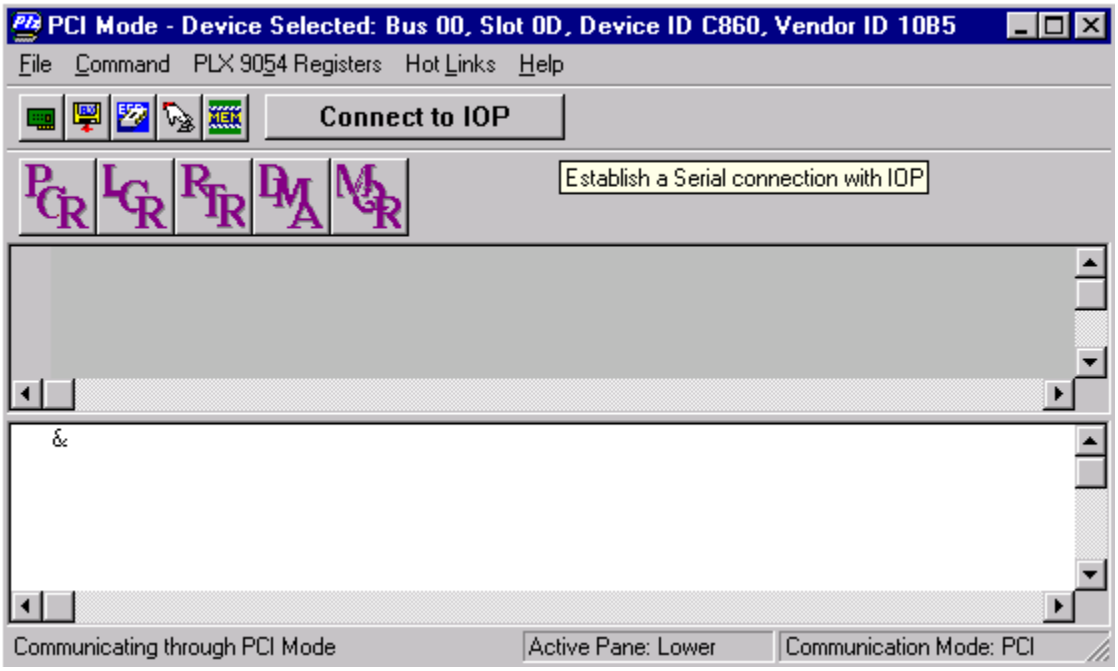

PCI Mode - Device Selected: Bus 00, Slot 0D, Device ID C860, Vendor ID 10B5
File Command PLX 9054 Registers Hot Links Help
Connect to IOP
Establish a Serial connection with IOP
PCR LCR PTR DMA MBR
&
Communicating through PCI Mode
Active Pane: Lower
Communication Mode: PCI

Figure 2-1. The PLXMon 2000 Interface

Depending on whether a supported PLX RDK is present, the monitor will enter PCI Mode or Local IOP (Serial) Mode. If a PLX RDK is in a PCI slot on your computer you will be in PCI mode, otherwise you will enter Local IOP (Serial) Mode. It is possible to switch back and forth between the two modes after the program has started. Also note that this program can be run on a second computer and may be used to do remote debugging through a serial port. PLXMon 2000 recognizes boards by checking the vendor and device IDs.

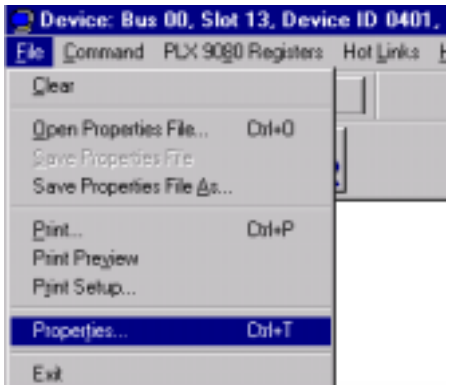


Figure 2-2. Selecting the Properties

If PLXMon 2000 does not detect a PLX RDK in the system, it will notify the user that one does not exist. If you are using your own board, you must add a new device to the Properties menu or verify that the device data is correct in the Properties menu. Until properties are assigned to the specific Vendor and Device ID of your

© PLX Technology, Inc. All rights reserved.

board, the program will not know the data needed to access the board. To edit these values select the Properties item under the File pull-down menu. (You can also use the hot-key Ctrl-t).

The Properties menu first shows the Device Configuration page. Figure 2-3 shows the Device Configuration screen for the IOP 480RDK. This data is used for PCI access only; the serial configuration data is retrieved using Back End Monitor commands directly through the serial port. The Properties data will be correct if you are using a PLX RDK board. For the IOP 480RDK board, under Local CPU Options, Big Endian CPU is selected by default because IOP 480 supports Big Endian.

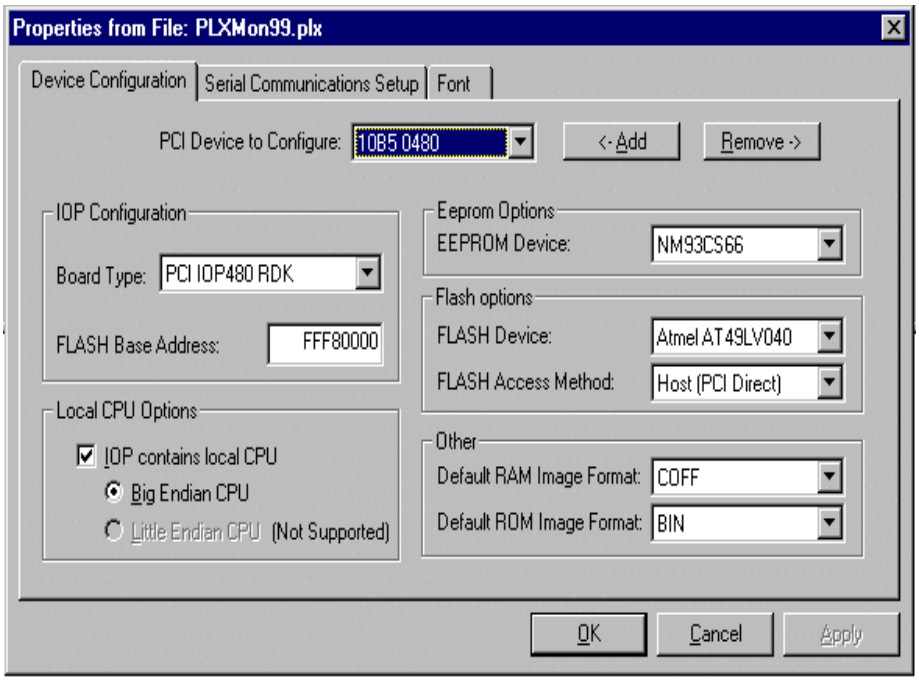
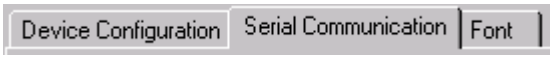


Figure 2-3. Device Configuration

These are the steps that should be followed to ensure proper operation of PLXMon 2000:

1. If the Vendor and Device ID are not listed under the “PCI Device to Configure” combo box, then click the “ADD” push button to create a new custom property entry for this device.
2. Now enter all the valid device information. By first clicking the RDK Default button, the default values for the RDK type selected will be entered on the screen. You should still verify that the Configuration EEPROM and FLASH EEPROM are the correct device types.
3. Change the memory map, or set default data file extension types, if necessary.

Now click the Serial Communications tab.



If you wish to use the debug port on your RDK, then you need to select the correct COM port. Also note if you are in Serial mode (entered by clicking on Connect to IOP) when doing this change, you must switch out, then back into Serial mode for your changes to take effect. The IOP software on the PLX RDKs has been designed to accept 38400 Baud only. You can also change the timeout length if desired.

To see command line data in a different font, select the font tab to select many different styles and point sizes. Select the underlined option and it will turn the display screen into lined pages.

© PLX Technology, Inc. All rights reserved.

2.1.1 Default Vendor and Device IDs for PLX Reference Design Boards

PLXMon 2000 supports a number of PLX Reference Design Boards. They are listed in Table 2-1 along with their vendor and device IDs for your reference.

Table 2-1. RDK Boards Supported by PLXMon 2000

Name of PLX Boards	Default Device ID	Default Vendor ID
IOP 480RDK	0480	10B5
PCI 9054RDK-860	1860	10B5
CompactPCI 9054RDK-860	C860	10B5
PCI 9054RDK-LITE	5406	10B5
PCI 9080RDK-860	0860	10B5
PCI 9080RDK-401B	0401	10B5
PCI 9030RDK-LITE	3001	10B5
CompactPCI 9030RDK-LITE	30C1	10B5

2.2 Displaying Registers

Whether you are in PCI mode or Serial mode, you can access the registers on a PLX RDK. It's as easy as clicking a button. The large buttons on the lower tier display various register sets. Click on the PCI Configuration Registers (PCR) button. A formatted PCI register set appears with some bit decoding already done. Grayed-out boxes indicate read-only windows. Now close this window and open the Local Configuration Register (LCR) window.

By clicking on an edit box, you can change the hexadecimal value. Then either close the window or move the cursor to another edit box to enter the value. Clicking on a check box will automatically update the register. Try it by clicking on the Details of any register box. The next dialog that appears will have check boxes to represent various bits of the register.

All the register capabilities mentioned above are applicable in Serial Mode as well as in PCI mode. PLXMon 2000 uses the interface provided by the Back End Monitor (BEM) to read and write to register locations. For more information on BEM, see either the Host SDK Programmers Manual or the glossary.

2.3 Popular Features

Now you can try using the built in applications that make PLXMon 2000 very useful. This section will give you a brief overview of the interactive downloader application as well as interactive DMA.

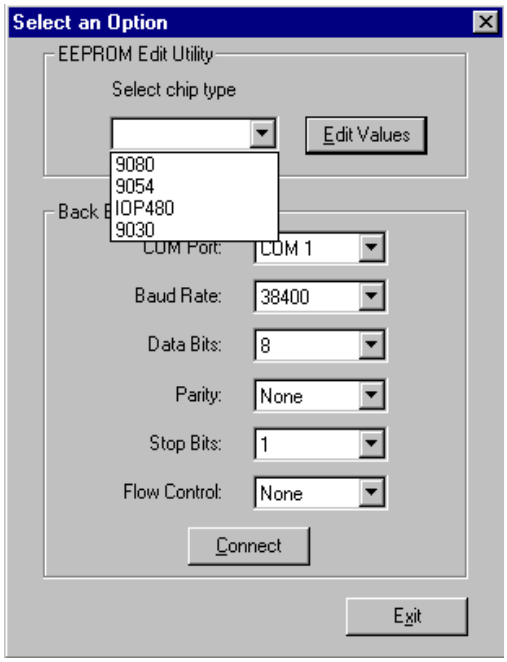


Figure 2-4. EEPROM Edit Utility

© PLX Technology, Inc. All rights reserved.

2.3.1 EEPROM Edit Utility

The EEPROM Edit Utility allows you to program a blank EEPROM. (Figure 2-4)

NOTE: THIS SCREEN IS ONLY AVAILABLE WHEN YOU START PLXMON 2000 WITH NO BOARD PLUGGED INTO YOUR PC.

You simply select the PLX chip you are using and click on <Edit Values> push button. This will take you to the chip specific EEPROM screen. All values on this screen are zero. On this screen, you can either enter your own values or load the values from a file. (The Host SDK contains a *.eep file for each PLX board in the …\hw\eeprom\… directory). Then, you can save this file to a floppy disk and walk over to an I/O programmer and program your blank EEPROM device.

2.3.2 Downloading To The IOP

Note: This feature is available with the PCI Pro SDK.

If you want to download an application to the IOP side, this is the feature you will need to use. Some of the features of the download utility include:

- Translation from different file formats including COFF, IBM ELF, Motorola S-Record, and pure Binary.
- The ability to download the file to either RAM or FLASH ROM.
- Binary reads from FLASH ROM to a binary data file.
- Supports Serial downloads to RAM.

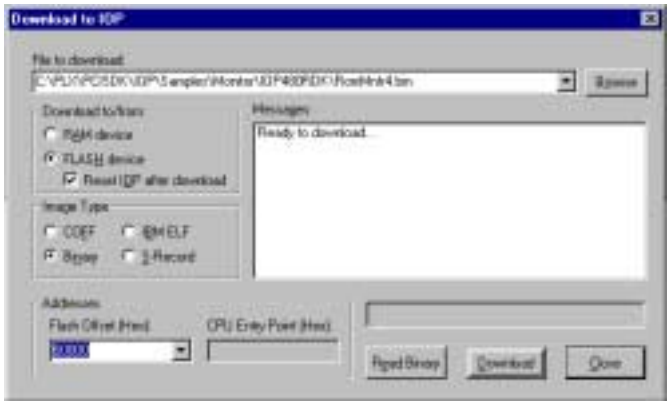


Figure 2-5. Download To IOP

To familiarize yourself with downloading an application/image to an RDK or to your target board, this tour will take you through the steps of downloading a RAM Hello Sample, and the Direct Master ROM. The examples demonstrate downloading using the PCI 9054RDK-860, but the methods also apply to other PLX RDKs.

Downloading to RAM

a. You must first select either a Serial or PCI channel before opening the Download window. If performing a PCI download, you can connect a serial port to another PC running PLXMon 2000, and see the download in progress.

b. Open the Download Window by clicking the download to embedded icon, which looks like a small disk with a downward facing arrow.

c. In the Device Configuration information, RAM data is stored in COFF format for the currently selected device. Therefore the default file type will be COFF.

1. Go to the directory:

   <Install-Path>\PLX\PciSdk300\IOP\Samples\Hello\9054RDK-860 and select the file RamHello.cof.

© PLX Technology, Inc. All rights reserved.

2. Click on download. By reading the status screen, you can verify that the program was successfully downloaded (See Figure 2-4).

Burning a FLASH ROM

Note: *If in the course of writing a program to the FLASH an error occurs, do not shut down the computer until you have re-burned a valid ROM program. Failure to do so will require removal of the FLASH ROM chip and re-burning in an external device programmer.*

1. Downloads to FLASH are supported via the PCI bus and Serial port. In this example, we will be downloading an image to the FLASH via the PCI bus. So, select the FLASH device.
2. After selecting FLASH, set the desired image type. It will also be necessary to give an offset from the physical FLASH address set in the configuration menu. This is necessary because certain RDKs produce FLASH data to be downloaded to different addresses. The following chart describes the offsets used for each RDK.

RDK Type	FLASH OFFSET (in HEX)
IOP 480RDK	60000
PCI 9054RDK-860	0
CompactPCI 9054RDK-860	0
PCI 9080RDK-860	0
PCI 9080RDK-401B	60000
PCI 9080RDK-RC32364	N/A
PCI 9080RDK-SH3	0
PCI 9054RDK-LITE	0

Valid FLASH Offsets

3. Go to the directory: <Install-Path>\PLX\PciSdk300\IOP\Samples\Dmaster\9054RDK-860 and select the file RomDM.bin.
4. Now click on download. Again you can verify that the download was successful by reading from the serial port on another PLXMon 2000, and by reading the Status window as well. A good way to test the ROM code is to reset the RDK, and verify the same program is running after the reset.

For a more complete description of the IOP download function consult the reference section.

2.3.3 DMA Transfers

DMA transfers can be used to transfer data rapidly between the IOP and the PCI bus. This PCI bus could be a user-mapped common buffer, or another RDK's local space window. This example will demonstrate how to set up a simple IOP to PCI DMA transfer.

1. To set up the DMA transfer, some registers must be properly initialized. First click on the DMA button, to display the registers to be modified. Set up the DMA transfer as shown in Figure 2-6.

2. At the & prompt, which is located in the Lower Pane, various commands can be entered to get information on the RDK board that is selected. The "VARS" command (started by typing "vars" at the & prompt) gives two values for Hbuf, a virtual and a physical address. The physical address is used for the PCI Address (Lower 32 bits). Enter a value for Transfer Count, 500 is used in this example. Entering dl s0+100000 shows you the current contents of the local memory on PLX RDK board at location 1MB. You can use the EL command for editing and writing to local memory.

© PLX Technology, Inc. All rights reserved.

3. The typical requirement for setting up the mode register is to set the Ready Input Enable and the bus width to 32 bits for PCI 9054 and PCI 9080 based RDKs (see figure 2-6). These two combinations give the Mode register a value of 43(h). For the IOP 480RDK, the Mode register value can be 0 (see figure 2-7). The Local Address given here is valid for all supported RDKs and is 100000(h). The Data Transfer direction is set within the Descriptor Pointer by clicking on the Details button, and setting the direction to "Local to PCI." The Threshold Register has a default value of 0.

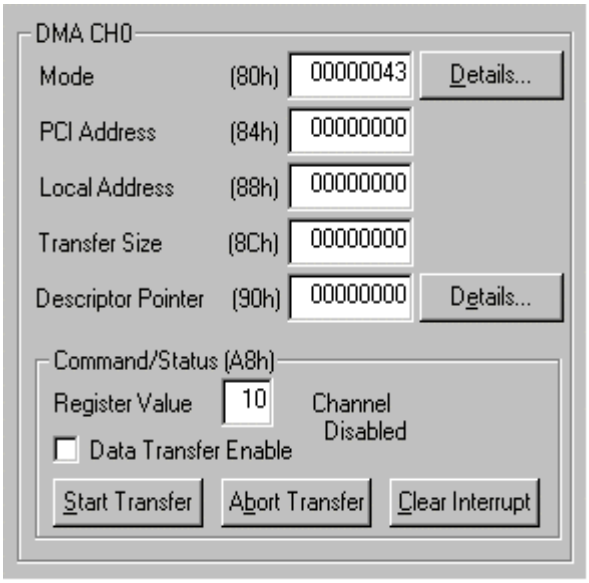


Figure 2-6. PCI 9080 and 9054 DMA Channel 0 Registers Channel 0 Registers

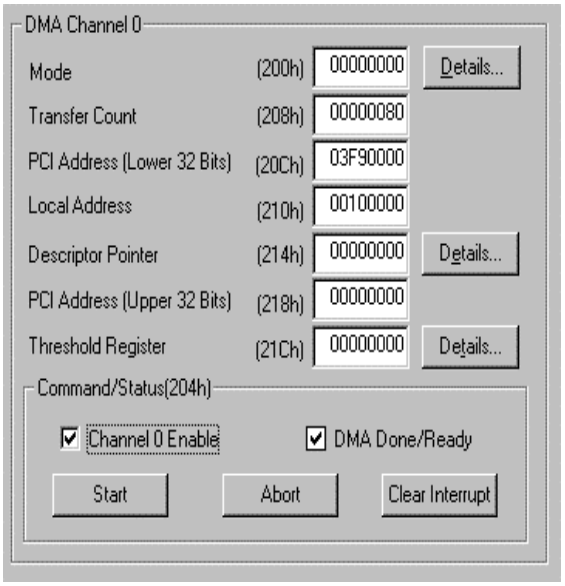


Figure 2-7. IOP 480 DMA Channel 0 Registers

4. To start the transfer, first enable the transfer by checking the Channel 0 Enable box. Clicking on Start button will begin the operation. The DMA Done/Ready bit will remain checked because the transfer count of 80 is very small and the transfer completes very quickly.
5. To verify that the data was transferred correctly, you can read the PCI buffer, Hbuf, by typing dl hbuf at the & prompt in the lower pane.

This concludes the tour of PLXMon 2000. More details about the specific PLX devices are included in the appendices.

2.4 Board with No Local CPU

If you are using the PCI 9054RDK-LITE, PCI 9030RDK-LITE or CompactPCI 9030RDK-LITE boards or if you are not using a local CPU on your board, then the following information may be helpful to you.

2.4.1 Using PCI XXXX RDK-LITE Board or Your Custom Board with No CPU

The PCI 9054RDK-LITE, PCI 9030RDK-LITE or CompactPCI 9030RDK-LITE board packages contain the PCI Host SDK CD-ROM. You can install the Host SDK by inserting this CD-ROM into your PC's CD-ROM drive. When you start PLXMon 2000, you will get the screen shown in Figure 2-8 below:

© PLX Technology, Inc. All rights reserved.

Figure 2-8. PLXMon 2000 Startup Screen

From the <File> pull down menu, click on <Properties> and go to the <Device Configuration page>.

If you are using the PLX PCI 9054RDK-LITE board, then from the <Board Type> pull down menu, you should select <PCI 9054RDK-LITE>. Figure 2-9 shows the settings for the PCI 9054RDK-LITE board.

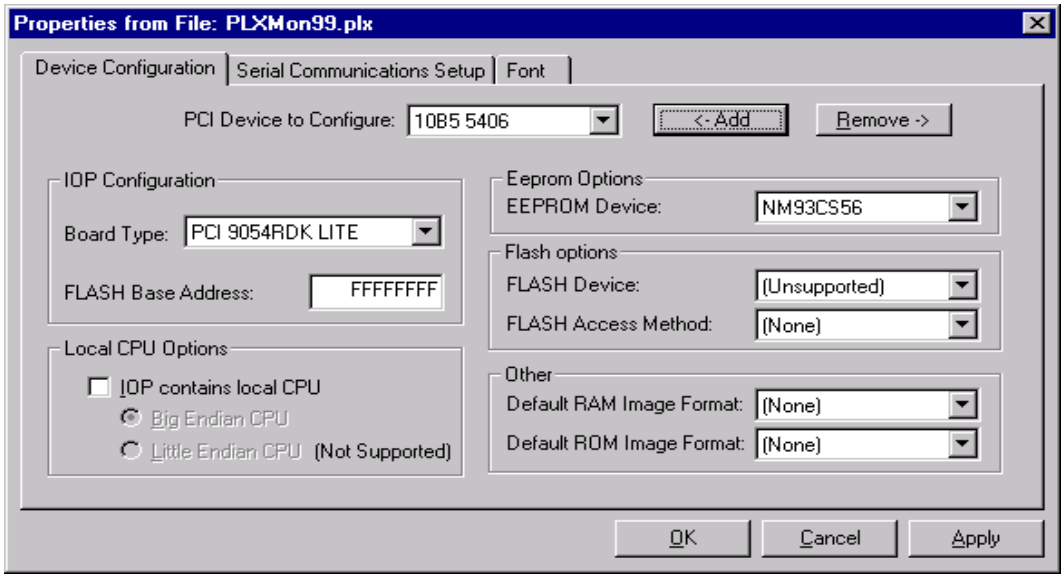


Figure 2-9. PCI 9054RDK-LITE Board Properties

© PLX Technology, Inc. All rights reserved.

If your board does not have a local CPU, you should de-select the <IOP contains local CPU> check box under the <Local CPU Options>. Figure 2-10 shows the Device Configuration screen for a custom board with a Device Id of ABCD and a Vendor Id of 10B5. These Ids were added by clicking on the <-Add> push down button. Next, you should configure some or all of the other options on this screen depending upon your board design.

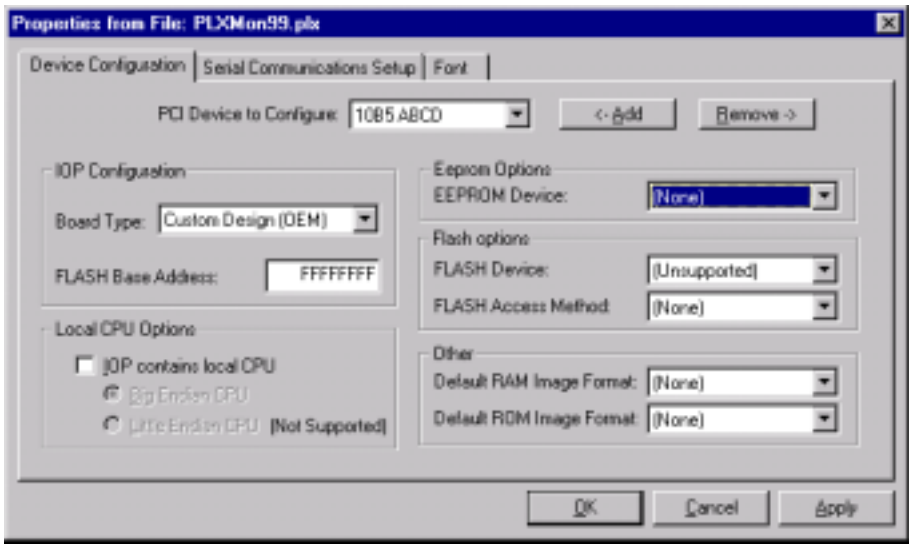


Figure 2-10. Properties for Board with No CPU

© PLX Technology, Inc. All rights reserved.

3. PLXMon 2000—Reference

The reference section is provided to give detailed information about every feature of PLXMon 2000. It is organized alphabetically by feature title.

3.1 Access Mode

The two methods of access to any RDK or custom board are via the PCI bus and via the Serial port. See Figure 3-1.

3.1.1 PCI Mode

When PCI mode is selected all communication between PLXMon 2000 and the PLX chip is done across the PCI bus, via the Host SDK API library and Windows device driver. This method will be familiar to users of previous PLXMon versions. The mode is selected by toggling the `Connect to IOP/Disconnect from IOP` button on the toolbar. In PCI mode, the "`Connect to IOP`" button is displayed. Essentially, the PCI mode results in all communication to the PLX device via its PCI Bus interface. All RDK properties used in PCI communication are found in the Device Configuration menu. Figure 3-1 shows the PLXMon 2000 interface in PCI mode.

3.1.2 IOP (Serial) Mode

Note: This feature is available with the PCI Pro SDK .

Clicking on the "Connect to IOP" toggle button enters IOP Mode. PLXMon 2000 automatically enters IOP mode if a PLX RDK is not present or if you are using your prototype board for the first time. When IOP (serial communication) mode is selected all communication between PLXMon 2000 and the PLX device by passes the Host SDK API library and device driver. Instead, PLXMon 2000 communicates with the IOP's BEM module (refer to Host SDK Programmer's Manual for BEM details). First, the program queries RDK information such as the PLX register base address. Essentially this results in all communication to the PLX device via its Local Bus interface. Commands are limited to local reads and writes, and a local reset.

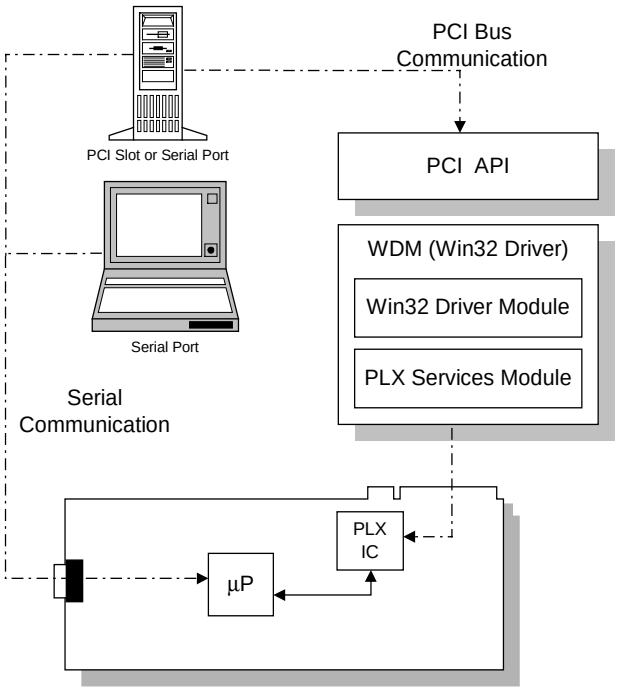


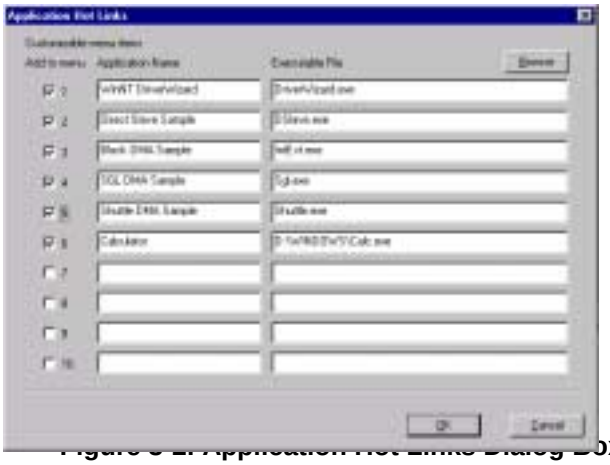
Figure 3-1. PCI vs. Serial Data Flow

Note: To use this mode, your IOP software must contain the BEM module. PLX RDKs, which have a local CPU, support this mode by default.

© PLX Technology, Inc. All rights reserved.

3.2 Application Hot Links

Selecting Hot Link from the PLXMon 2000 pull down menu bar can start the application hot links. The PLXMon 2000 offers the capability for users to add hot links to applications. Typical applications that a user may want to add hot links to are Host SDK samples, PLX manufacturing test software, or custom applications. Please see Figure 3-2.



To use this feature you need to enter the Application Name and associated path to the executable in the Hot Links pull-down menu. They will then appear in the Hot Link pull down menu.

3.3 Device Configuration

This dialog window can be found by pulling down the File menu then selecting Properties. PLXMon 2000 retains program options for specific PLX RDKs and custom boards. These options are used throughout PLXMon 2000 to setup default values for various program options. For example the loader uses the flash properties to determine the flash programming method needed.

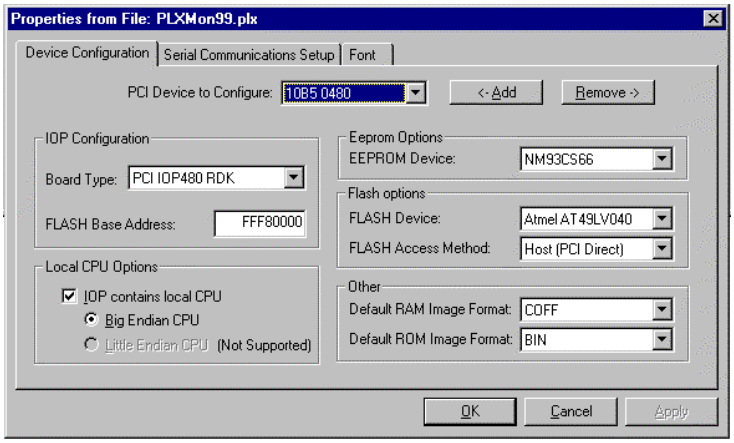


Figure 3-3. Device Configuration Dialog Box

All supported device attributes are saved to a **Properties File**. This file has a default name PLXMon 2000.plx and is automatically loaded. PLXMon 2000 will only search the current directory for this file so if the executable is moved, this file should be moved as well.

PLXMon 2000 uses the Vendor ID and Device ID to recognize a **supported** device. By adding a new entry to this list, a custom ID combination can be supported. Figure 3-3 shows the settings for the IOP 480RDK board. Similar settings must be selected for each RDK when PLXMon 2000 is used.

When adding a new device, first enter the custom Vendor ID and Device ID. Then select its RDK `type` (if applicable). On hitting RDK `Default`, the normal settings for that RDK will be entered. Press Apply or OK to enter the data into the Properties file.

© PLX Technology, Inc. All rights reserved.

3.4 Downloading IOP Applications

Note: This feature is available with the PCI Pro SDK.

In order to download to and execute programs on the IOP, the Download to IOP command is provided. This utility can be run by clicking on the icon, or by selecting it under the Command pull-down menu.

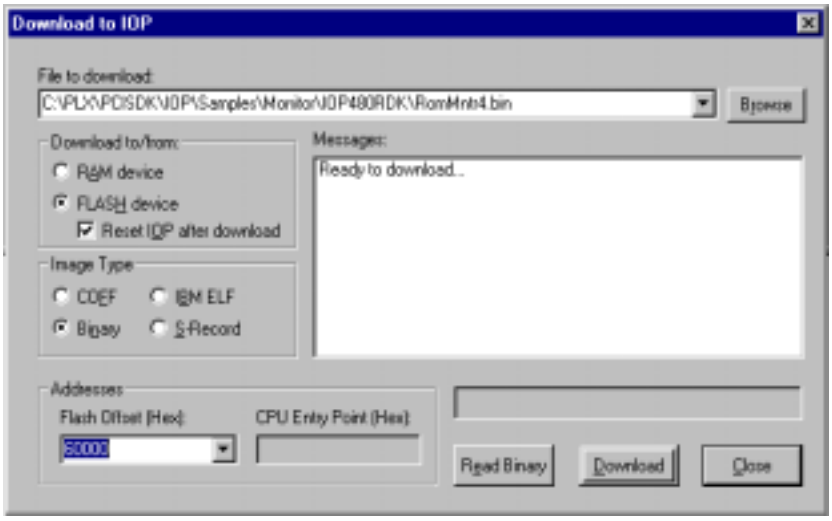


Figure 3-4. File Download Dialog Box

Files can be sent to either a RAM device or FLASH device and the base address is programmable.

Typically, all default values that are already selected will be correct for the download you wish to do. By selecting either RAM or FLASH, the file format will automatically be changed to the format that is specified for this device in the "Device Config" menu. These features can be overridden, and by doing so the user should be knowledgeable about the format of these files and the memory map of the RDK being programmed.

Note: Programming the FLASH can be dangerous and it is important that the setup parameters be correct before the download is attempted. Data like RDK Type, Flash Address, Programming Method, and Memory Offset must be known beforehand.

To read the data on a FLASH device into a file, the Read Binary button can be used in conjunction with the memory offset window. This data will be retrieved unformatted and stored as a pure binary file. Currently, the binary read function will create an image that starts at the memory offset provided and ends at the end of the usable flash range.

This utility also has the ability to program a device through the serial debug port. The serial download supports both ROM and RAM programming.

© PLX Technology, Inc. All rights reserved.

3.5 Font Configuration

This dialog screen can be reached from the Properties option in the File pull-down menu. The font and point size will be changed in the display screen only. The appearance of the data in the dialog boxes will not be changed.

Figure 3-5. Font Select Dialog Box

3.6 The Interface

The PLXMon 2000 main interface, shown in Figure 3-6 contains:

- A drop-down menu bar, with the five main drop-down menus. Certain options will be available, depending on which PLX RDK is selected. The PLXMon 2000 interface for the 9054RDK-860 is shown in Figure 3-6.
- A split screen interface used for simultaneous Command Line Interface (Lower Pane) and Serial Access (Upper Pane). Use the F6 key to toggle between active panes.
- An optional toolbar
- The status bar which reports the configuration file being used

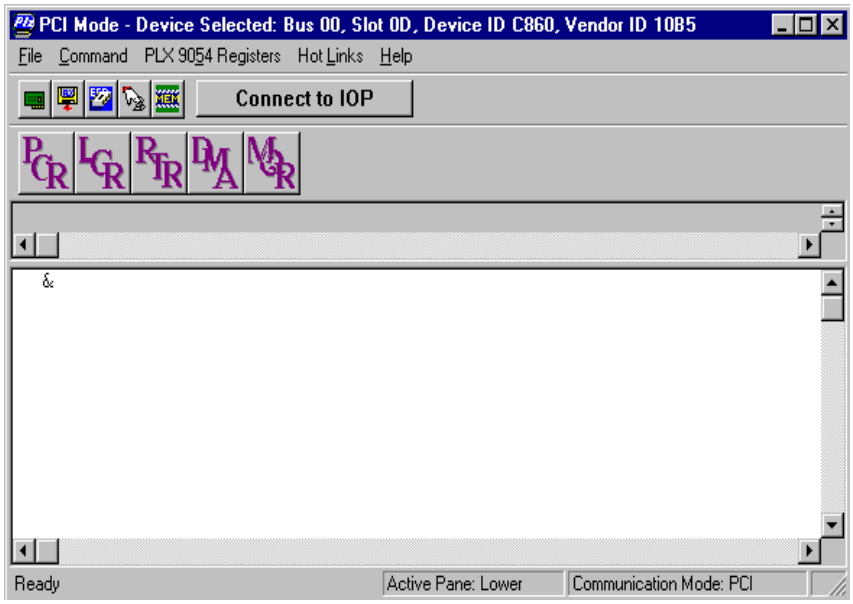


Figure 3-6. The PLXMon 2000 Interface

- The "Connect to IOP/Disconnect from IOP" button toggles between PCI and serial communication mode.

© PLX Technology, Inc. All rights reserved.

3.6.1 The PLXMon 2000 Toolbar

The PLXMon 2000 toolbar serves as an optional shortcut to the drop-down menu commands.

3.6.2 Status Bar

The status bar provides simple and useful tips. When the mouse is pointing on an object or a button in the main window of PLXMon 2000, the status bar displays information about it. Tool-tips are also displayed when the mouse pointer is held over an object for a short period.

3.6.3 Command Line Interface (CLI)

At the & prompt which is located in the Lower Pane, various commands can be entered to get information on the RDK that is selected. This command line can be used in both PCI and Serial modes. To get a list of valid commands at any time in PLXMon 2000, at the command line, type help or ?. The following sections will describe all the valid commands.

Note: The CLI is not case sensitive.

&help
Help is available for the following commands:
dl dw db el ew eb quit
? help r il iw ib vars
ol ow ob war rpw pci
For more information, type HELP command
&
Ready
Active Pane: Lower Pane
Communication Mode: PCI
PLXMon32.PLX

Figure 3-7. Command Line Interface Commands

3.6.3.1 Displaying Memory Via Memory Cycles, (dl, dw, db)

These commands display different sizes of data that are accessed through memory cycles. Using `dl` will return a 32-bit value, `dw` will return a 16-bit value, and `db` will return a 8-bit value. They follow the format:

```
<command> <address> [[l] bytelength]
```

By default the byte length is 80 [hex] bytes. Typing the command again with no arguments will make PLXMon 2000 continue to display the range with the same `bytelength` as before.

3.6.3.2 Displaying Memory Via I/O Cycles, (il, iw, ib)

The iX commands are similar in syntax with the dX commands except they access memory using I/O cycles instead of memory cycles. They follow the format:

```
<command> <address> [[l] bytelength]
```

By default, the bytelength is dependent on the command. Using `il` will return 32 bits of data, `iw` will return 16 bits, and `ib` will return 1 byte. All these lengths can be overridden, however. By re-typing the command with no arguments, PLXMon 2000 will continue to display the memory locations using the size of the data retrieved as the increment size.

3.6.3.3 Writing Memory Via Memory Cycles, (el, ew, eb)

Again the syntax of the write using memory cycles is similar to the `dX` commands. Using `el` will write values as 32 bit wide objects, `ew` will write 16 bit values, and `eb` will write one byte at a time.

They follow the format:

```
<command> <address> [INC increment] [value]
```

While the input address is required, both the value and the increment parameter are optional. By not entering the [value] parameter, the program will query you for data in interactive mode.

© PLX Technology, Inc. All rights reserved.

Interactive mode allows the user to press the space bar between typing the value he/she wants to enter and PLXMon 2000 will auto-increment the write address the size of the data being entered. Press the enter key to exit the command. By changing the `INC` parameter in the command line above, the auto-increment value can be changed. What follows is a brief example of how to use this command.

If the user types the command `eb s0 INC 4`, they wish to interactively write bytes to the location `s0` (which is a system label, more on this in section 3.7.3.6). Every time the user enters a value and hits space the program will query for the previous address plus INC, which is 4. On display of the memory range the user should see the following screen.

The 00: bytes that appear before each user entry are the previous values that are about to be overwritten by the user's data.

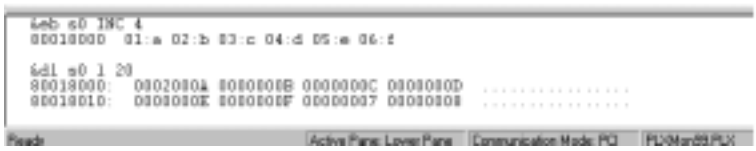


Figure 3-8. Interactive Mode

3.6.3.4 Writing Memory Via I/O Cycles, (ol, ow, ob)

The oX commands are simpler syntactically than the memory cycle writes. Each command writes a different sized data object to a port address. They require only two parameters:
`<command> <address> <value>`

3.6.3.5 The Pci Command

This command allows read/write access to the PCI configuration registers. Its syntax is as follows:

```
pci <pci offset> [value]
```

To write to the required offset, just add the value to write; otherwise the value will be displayed as a 32-bit register value.

Note: the offset will be different depending on the Access mode: PCI or Serial.

3.6.3.6 The Quit Command

The `quit` command terminates the application PLXMon 2000.

3.6.3.7 The Reg Command

The `reg` command allows users access to the PLX chips local register sets. Data can be read or written in 32 bit sizes at a given byte boundary. The syntax of the command is as follows:
`reg <register offset> [value]`

If a `[value]` is given, the command will write the data to the specified address.

3.6.3.8 The Repeat Command (r)

The repeat command can is used to make PLXMon 2000 repeat the command types before the `r` a set number of times. Its syntax is as follows:
`[command] r [iterations]`

© PLX Technology, Inc. All rights reserved.

If the number of `iterations` is not given, then PLXMon 2000 will execute the command indefinitely until the user hits a key. The command to be repeated must be in the same expression as the `r` command.

3.6.3.9 User Variables (vars)

PLXMon 2000 creates some user labels as a mnemonic aid for common memory locations. These strings can be used interchangeably with the values they represent.

This table lists the variables set by PLXMon 2000 and what memory ranges they represent.

Variable Name	Description	PCI or Serial
Plx	Plx register address	PCI
hbuf	User mapped region of the PCI common buffer.	PCI
s0	Local Space 0	PCI
s1	Local Space 1	PCI
s2	Local Space 2	PCI
s3	Local Space 3	PCI

Figure 3-9. User Variables and their definitions

3.6.3.10 The Ver Command

Displays the version data contained in the Host SDK software release. This version of PLXMon 2000 is compatible only with Host SDK v3.1. This command is usable only during PCI mode.

3.7 Print, Print Preview, and Print Setup

The print commands (`Print, Print Preview, Print Setup...`) can be found under the File pull-down menu. These selections enable the user to create a formatted picture of the display window. Print preview will allow you to see the formatted screen before you print.

3.8 Register Access/Register Sets

The contents of registers can be represented in one of two ways in PLXMon 2000. Usually when a full 32-bit register is being displayed, it is shown in an edit box in hexadecimal format *(the 0x prefix is implied)*. Typing in new values, if not grayed-out, can modify these boxes. The value(s) will be updated when the user closes the window or when the cursor is moved to another edit box.

Check boxes are also used to display and change individual bits in a register. If the check box is on a main dialog window then a state change is immediate. If the check box is in a Details dialog with other checkboxes, then the changes are made only upon closing the dialog box.

Note: Please refer to Appendices A and B for specific details on the IOP 480, PCI 9080, PCI 9054, and PCI 9030 device registers, respectively.

3.9 The Reset Button

Found on the taskbar, the reset button signals the PLX RDK that is currently selected to reset itself. This action can be taken during both PCI and Serial modes.

Note: The method used to reset the board is customized to work with the PLX RDKs. This feature should not be used on devices/boards that do not support the reset algorithm. Consult the BSP source code (included with the SDK Pro) for complete information on the specific algorithm(s) used with various PLX RDK boards.

© PLX Technology, Inc. All rights reserved.

3.10 Selecting Devices

The `Select A Device` menu item, shown in Figure 3-11, is used to select a PCI device that you want to access. The pointer points to the currently selected device.

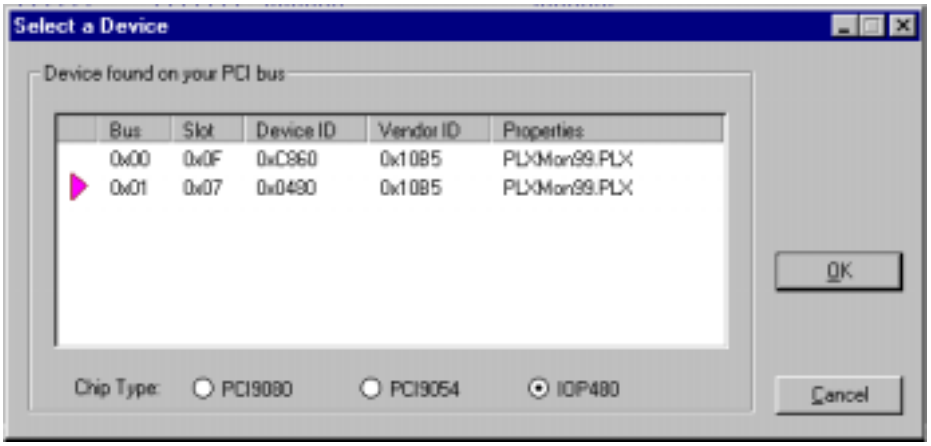


Figure 3-10. Device Select Dialog Box

Figure 3-10 lists two boards, the CompactPCI 9054RDK-860 (Device ID=0xC860) and the IOP 480RDK (Device ID=0x0480). To select a new device, move the highlight to the desired PCI device by using either the mouse or the cursor arrow keys, and click the OK button or press the Enter key. The pointer will not move to the new selection until OK button is pressed or until the item is double-clicked. The chip type radio button indicates which PLX device is present on the selected device.

3.11 Serial Configuration

PLXMon 2000 offers the capability to communicate with the PLX device through the serial port. To do so you must configure the appropriate serial port settings as shown in the figure below.

Communication ports COM1 through COM4 are supported. Baud rates 9600, 19200, 38400, and 57600 are supported.

Note: All PLX RDKs should be configured to 38400 baud, 8 Data Bits, No Parity, 1 Stop Bit, and no Flow Control.

Figure 3-11. Serial Communications Properties Dialog Box

© PLX Technology, Inc. All rights reserved.

3.12 Serial EEPROM Access

Pressing the Serial EEPROM button on the toolbar of PLXMon 2000 will start the EEPROM dialog box. The EEPROM screen for IOP 480RDK is shown in Figure 3-12:

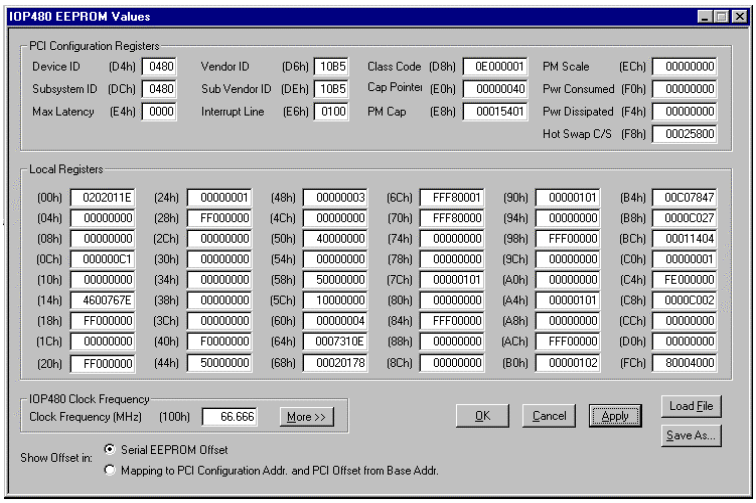


Figure 3-12. Serial EEPROM Access Screen

PLXMon 2000 supports saving the EEPROM values to a file and also loading values from a file.

Note: Having the wrong Serial EEPROM type selected in the Device Config. Menu can cause PLXMon 2000 to read/write invalid data.

3.13 Memory Access

Pressing the Memory button on the toolbar of PLXMon 2000 will start the Memory Display dialog box. The dialog screen of the CompactPCI 9054RDK-860 is shown in Figure 3-13: Memory Display supports both PCI and IOP memory read or write.

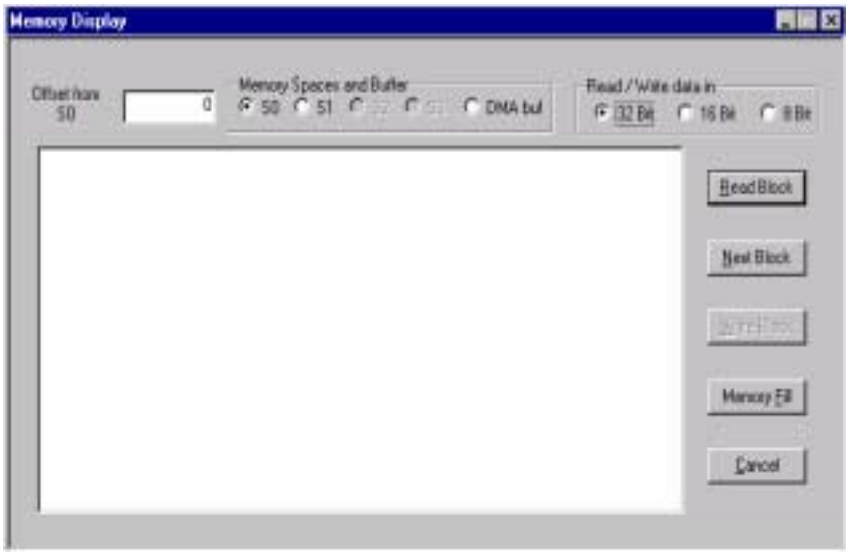


Figure 3-13. Memory Display Screen

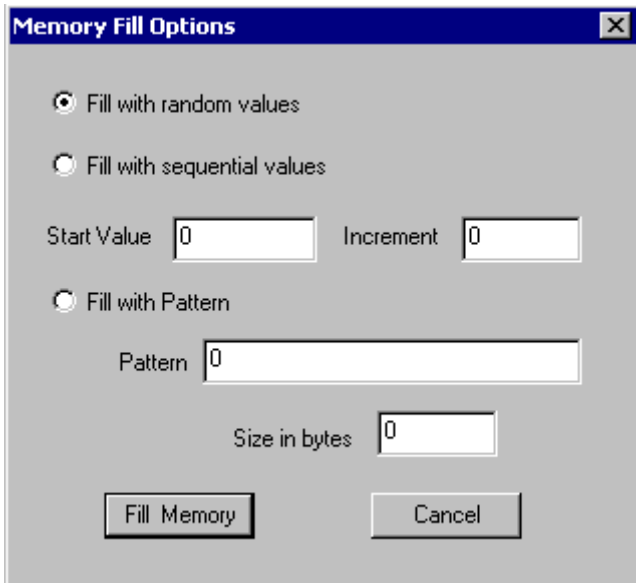
In PCI mode, clicking the "Read Block" button will read 0x100 bytes from the indicated offset address of the specified memory space in the specified bus width and display the data with corresponding Windows virtual addresses. Clicking the "Write Block" button will write the current displaying data to user specified offset address.

© PLX Technology, Inc. All rights reserved.

In IOP mode, clicking the “Read Block” button will read 0x100 bytes from the user specified local address in specified bus width and display the data with corresponding addresses (absolute local memory addresses). Clicking the “Write Block” button will write the current displaying data to user specified local address.

Note: Giving the wrong offset or local address can cause PLXMon 2000 to read/write invalid memory areas. **IOP mode memory access is available with the PCI Pro SDK.**

The Memory Fill function enables the user to write memory locations with a specified data pattern. Click, “Memory Fill” button opens the Memory Fill Options dialog box as shown in Figure 3-14:



Select the type of values and fill the edit boxes if necessary. Give the size of memory locations to be filled in number of bytes. Click ‘Fill Memory’ to write to user specified memory locations.

Figure 3-14. Memory Fill Options dialog box

The example below shows the settings of writing to user defined locations with a user defined data pattern.

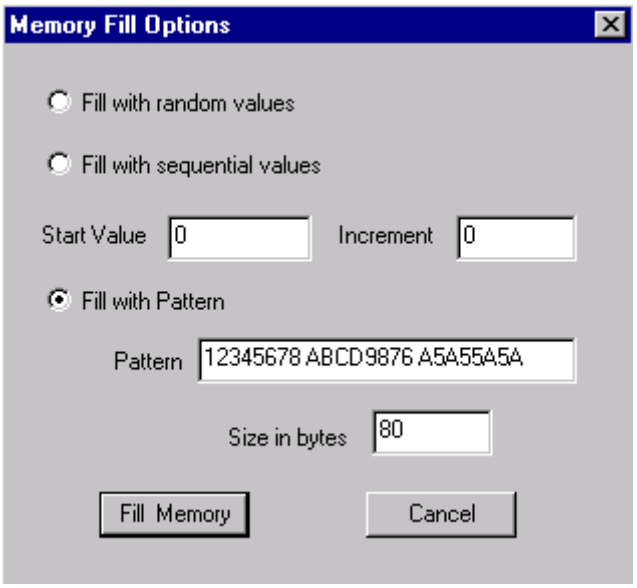


Figure 3-15. User defined data pattern

© PLX Technology, Inc. All rights reserved.

When editing the data pattern, a space should be given between the values. All the edit boxes accept hexadecimal values only. Please see Figure 3-16 for the memory display of above Memory Fill Options setting.

Note: Giving the wrong memory size in bytes to write can cause PLXMon 2000 to write invalid memory areas.

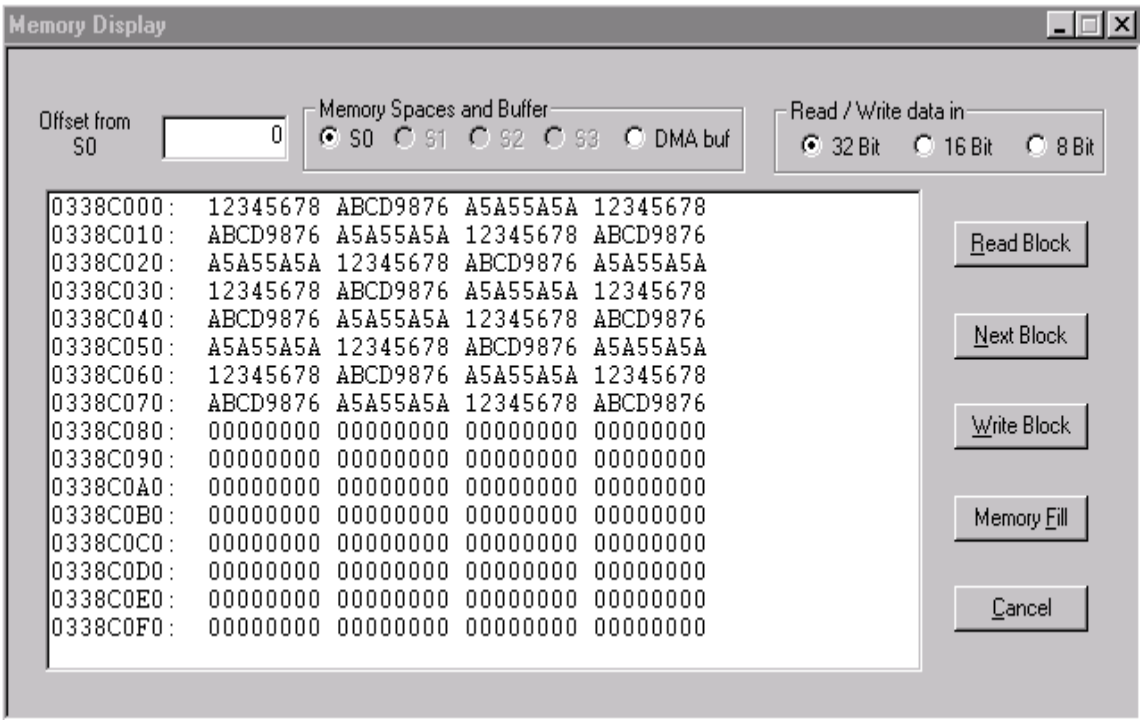


Figure 3-16. Memory filled with user defined data pattern

© PLX Technology, Inc. All rights reserved.

4. The IOP 480 Register Set

Each of the IOP 480 register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base address, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers if necessary.

4.1 The Register Group Dialog Boxes

The PLXMon 2000 toolbar for the IOP 480 contains seven buttons for register accesses. They are `PCI Configuration Registers (PCR)`, `Local Configuration Registers (LCR)`, `RunTime Registers (RTR)`, `DMA Registers (DMA)`, `Messaging Queue Registers (MQR)`, `Memory Controller Registers (MCR)`, and `IOP 480CPU registers (IOP480 CPU)`.

4.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in Figure 4-1, in the `PCI Configuration Registers` dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and checkboxes indicate the current settings of the register bit fields. To update the contents of the dialog box push the `Refresh` button.

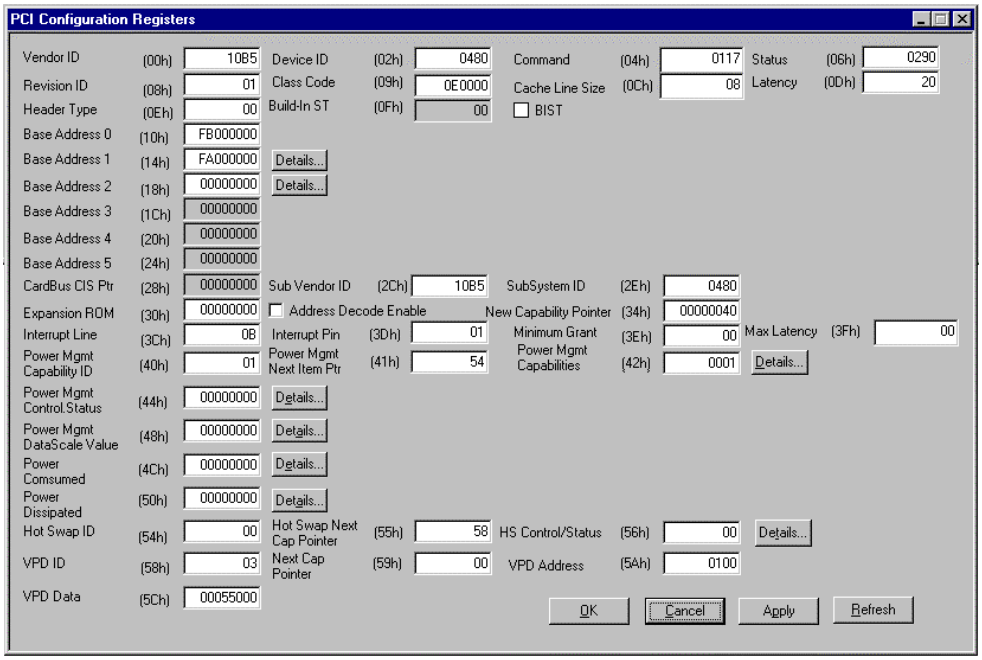


Figure 4-1 PCI Configuration Register Group Dialog Box

© PLX Technology, Inc. All rights reserved.

4.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes, in Figure 4-2, respectively. The `size` text box reflects the value (in bytes) of the associated register. The memory size is calculated from the corresponding register value and cannot be modified directly. To change the memory size, modify the associated register.

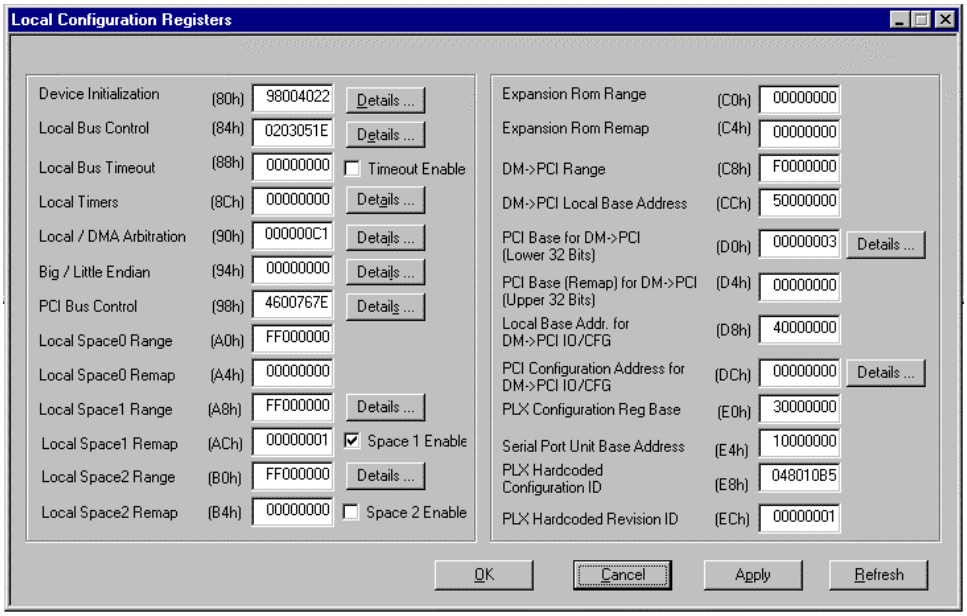

Local Configuration Registers
Device Initialization (80h) 98004022 Details ...
Local Bus Control (84h) 0203051E Details ...
Local Bus Timeout (88h) 00000000 Timeout Enable
Local Timers (8Ch) 00000000 Details ...
Local / DMA Arbitration (90h) 000000C1 Details ...
Big / Little Endian (94h) 00000000 Details ...
PCI Bus Control (98h) 4600767E Details ...
Local Space0 Range (A0h) FF000000
Local Space0 Remap (A4h) 00000000
Local Space1 Range (A8h) FF000000 Details ...
Local Space1 Remap (ACh) 00000001 Space 1 Enable
Local Space2 Range (B0h) FF000000 Details ...
Local Space2 Remap (B4h) 00000000 Space 2 Enable
Expansion Rom Range (C0h) 00000000
Expansion Rom Remap (C4h) 00000000
DM->PCI Range (C8h) F0000000
DM->PCI Local Base Address (CCh) 50000000
PCI Base for DM->PCI (Lower 32 Bits) (D0h) 00000003 Details ...
PCI Base (Remap) for DM->PCI (Upper 32 Bits) (D4h) 00000000
Local Base Addr. for DM->PCI IO/CFG (D8h) 40000000
PCI Configuration Address for DM->PCI IO/CFG (DCh) 00000000 Details ...
PLX Configuration Reg Base (E0h) 30000000
Serial Port Unit Base Address (E4h) 10000000
PLX Hardcoded Configuration ID (E8h) 048010B5
PLX Hardcoded Revision ID (ECh) 00000001
OK
Cancel
Apply
Refresh

Figure 4-2. Local Configuration Registers Dialog Box

4.1.3 The Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays, and allows modification of, the current register values of the Runtime Registers. See figure below:

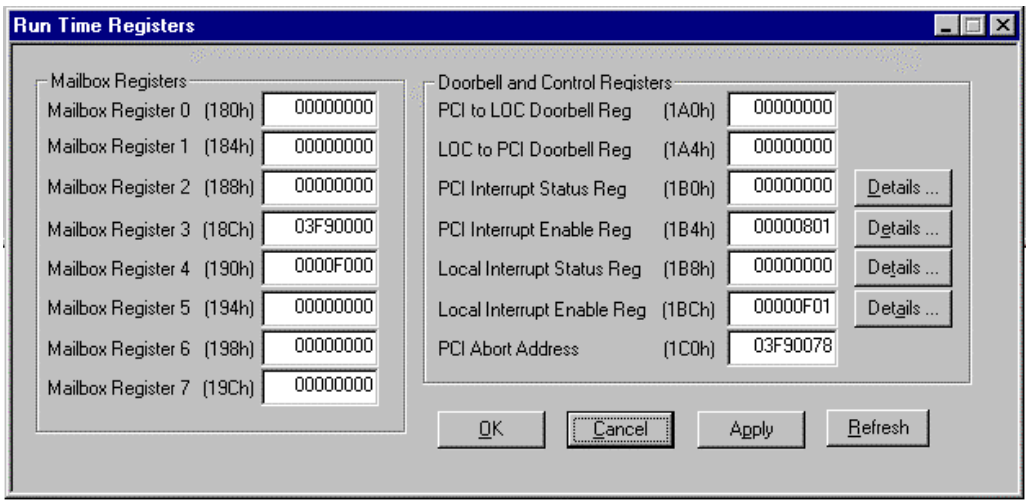

Run Time Registers
Mailbox Registers
Mailbox Register 0 (180h) 00000000
Mailbox Register 1 (184h) 00000000
Mailbox Register 2 (188h) 00000000
Mailbox Register 3 (18Ch) 03F90000
Mailbox Register 4 (190h) 0000F000
Mailbox Register 5 (194h) 00000000
Mailbox Register 6 (198h) 00000000
Mailbox Register 7 (19Ch) 00000000
Doorbell and Control Registers
PCI to LOC Doorbell Reg (1A0h) 00000000
LOC to PCI Doorbell Reg (1A4h) 00000000
PCI Interrupt Status Reg (1B0h) 00000000 Details ...
PCI Interrupt Enable Reg (1B4h) 00000801 Details ...
Local Interrupt Status Reg (1B8h) 00000000 Details ...
Local Interrupt Enable Reg (1BCh) 00000F01 Details ...
PCI Abort Address (1C0h) 03F90078
OK
Cancel
Apply
Refresh

Figure 4-3. Runtime Register Group Dialog Box

© PLX Technology, Inc. All rights reserved.

The Interrupt Enable/Status Register Dialog Box

There are four Details buttons for Interrupt Enable/Status Registers. These are for PCI and Local Interrupt Status and Enable Registers. The Interrupt Enable/Status Register Dialog Boxes provide information on the current value of the Interrupt Enable/Status register. The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

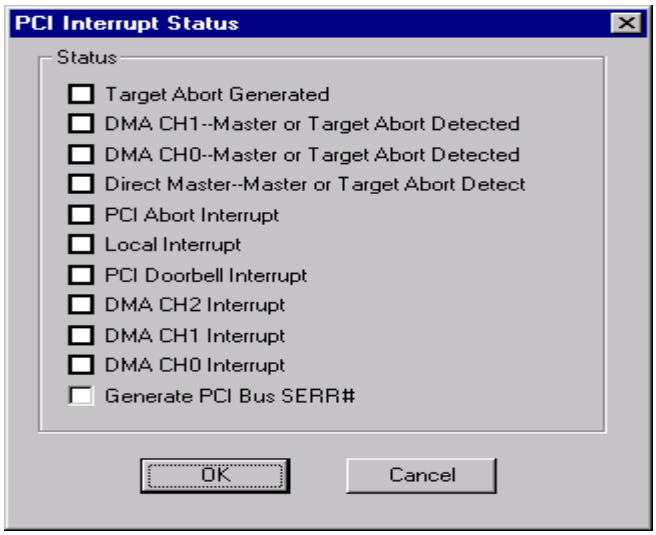


Figure 4-4. Interrupt Enable /Status Register Dialog Box

4.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for all DMA channels. See figure below.

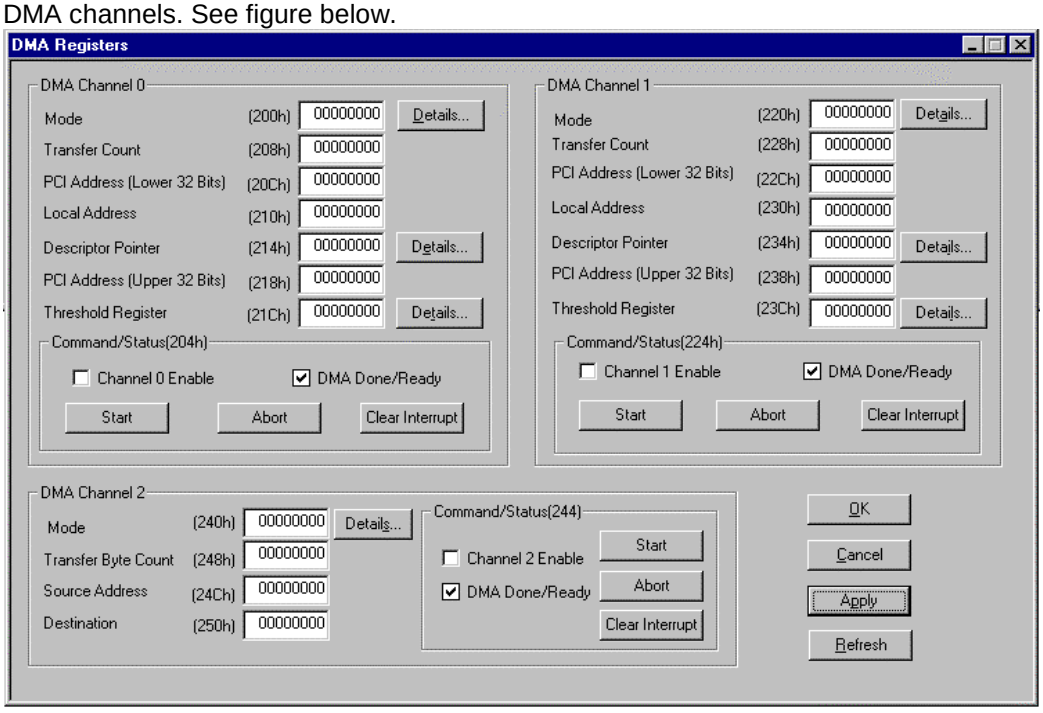


Figure 4-5. DMA Register Group Dialog Box

The Start and Abort Transfer buttons initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Clear Interrupt button resets the interrupts to their default state. The Channel Enable bit enables DMA transfers and

© PLX Technology, Inc. All rights reserved.

activates the Start, Abort, and Clear Interrupt buttons. The DMA Done/Ready bit indicates the DMA engine status.

4.1.5 The Messaging Queue Register Group Dialog Box

The Messaging Queue Register Group dialog box contains the current values for the Messaging FIFO Registers as shown below:

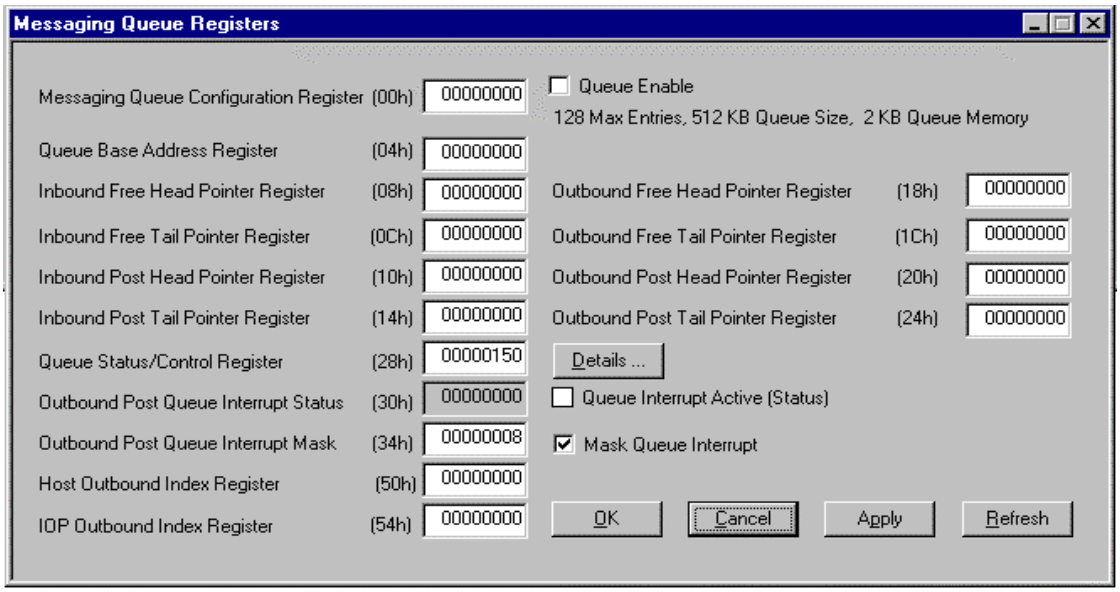

Messaging Queue Registers
Messaging Queue Configuration Register (00h) 00000000
Queue Enable
128 Max Entries, 512 KB Queue Size, 2 KB Queue Memory
Queue Base Address Register (04h) 00000000
Inbound Free Head Pointer Register (08h) 00000000
Outbound Free Head Pointer Register (18h) 00000000
Inbound Free Tail Pointer Register (0Ch) 00000000
Outbound Free Tail Pointer Register (1Ch) 00000000
Inbound Post Head Pointer Register (10h) 00000000
Outbound Post Head Pointer Register (20h) 00000000
Inbound Post Tail Pointer Register (14h) 00000000
Outbound Post Tail Pointer Register (24h) 00000000
Queue Status/Control Register (28h) 00000150
Details ...
Outbound Post Queue Interrupt Status (30h) 00000000
Queue Interrupt Active (Status)
Outbound Post Queue Interrupt Mask (34h) 00000008
Mask Queue Interrupt
Host Outbound Index Register (50h) 00000000
IOP Outbound Index Register (54h) 00000000
OK
Cancel
Apply
Refresh

Figure 4-6. Messaging Queue Register Group

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified, as this is a hardware-generated interrupt.

© PLX Technology, Inc. All rights reserved.

4.1.6 IOP 480 Memory Controller register Group Dialog Box

The Memory Controller Register Group dialog box contains the current values for the IOP 480's integrated memory controller. The memory controller register group is divided into five subgroups (see figure below), namely, LCS0, LCS1, LCS2, LCS3, and DRAM Control registers. For more information, please consult the IOP 480 data book.

Note: (LCS = Local Chip Select)

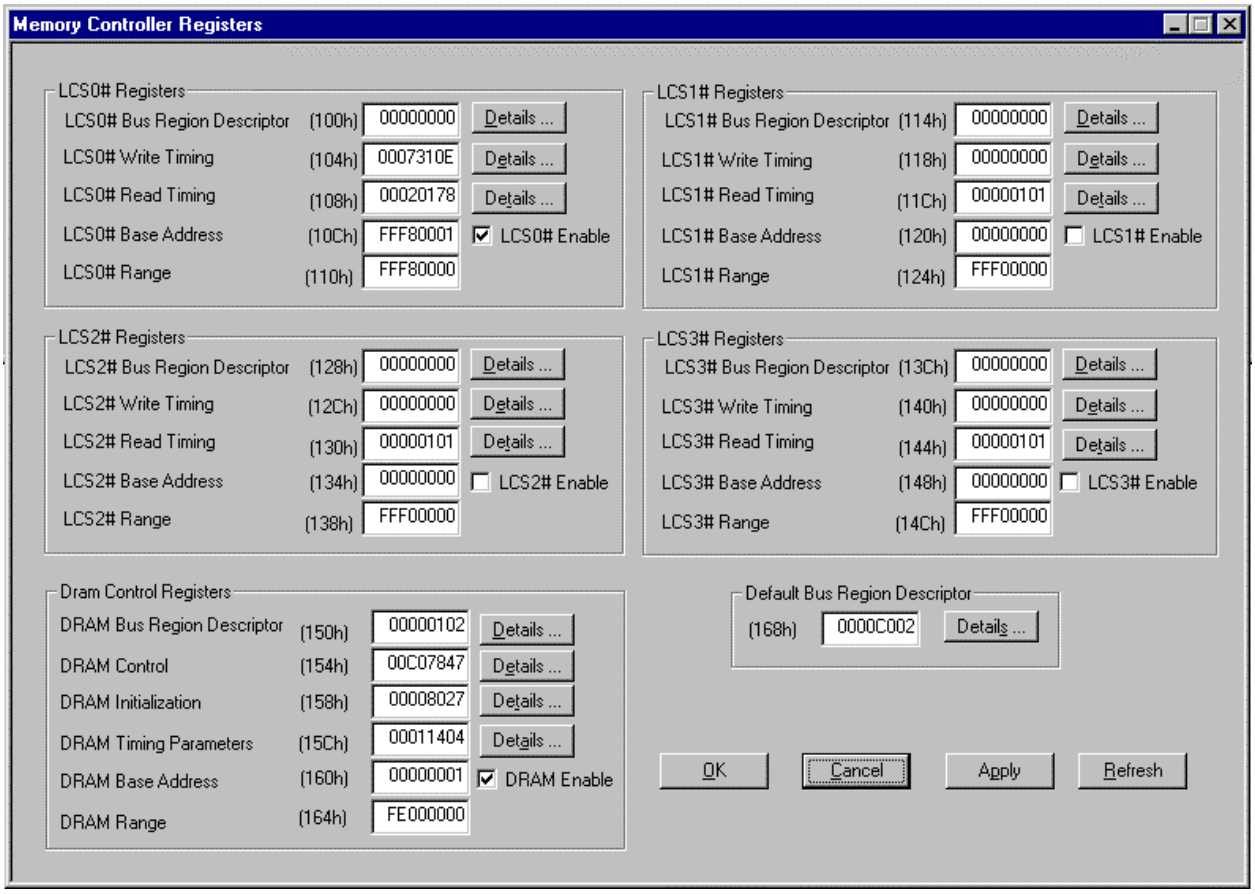

Memory Controller Registers
LCS0# Registers
LCS0# Bus Region Descriptor (100h) 00000000 Details ...
LCS0# Write Timing (104h) 0007310E Details ...
LCS0# Read Timing (108h) 00020178 Details ...
LCS0# Base Address (10Ch) FFF80001 LCS0# Enable
LCS0# Range (110h) FFF80000
LCS1# Registers
LCS1# Bus Region Descriptor (114h) 00000000 Details ...
LCS1# Write Timing (118h) 00000000 Details ...
LCS1# Read Timing (11Ch) 00000101 Details ...
LCS1# Base Address (120h) 00000000 LCS1# Enable
LCS1# Range (124h) FFF00000
LCS2# Registers
LCS2# Bus Region Descriptor (128h) 00000000 Details ...
LCS2# Write Timing (12Ch) 00000000 Details ...
LCS2# Read Timing (130h) 00000101 Details ...
LCS2# Base Address (134h) 00000000 LCS2# Enable
LCS2# Range (138h) FFF00000
LCS3# Registers
LCS3# Bus Region Descriptor (13Ch) 00000000 Details ...
LCS3# Write Timing (140h) 00000000 Details ...
LCS3# Read Timing (144h) 00000101 Details ...
LCS3# Base Address (148h) 00000000 LCS3# Enable
LCS3# Range (14Ch) FFF00000
Dram Control Registers
DRAM Bus Region Descriptor (150h) 00000102 Details ...
DRAM Control (154h) 00C07847 Details ...
DRAM Initialization (158h) 00008027 Details ...
DRAM Timing Parameters (15Ch) 00011404 Details ...
DRAM Base Address (160h) 00000001 DRAM Enable
DRAM Range (164h) FE000000
Default Bus Region Descriptor
(168h) 0000C002 Details ...
OK
Cancel
Apply
Refresh

Figure 4-7. IOP 480 Memory Controller Registers

© PLX Technology, Inc. All rights reserved.

4.1.7 IOP 480 CPU Registers Group Dialog Box

The IOP 480 CPU Register Group dialog box contains the current values for the IOP 480 CPU registers. These are divided into two groups, General Purpose Registers and Special Purpose Registers. Please refer to the IOP 480 data book for more information on these registers. See figure below:

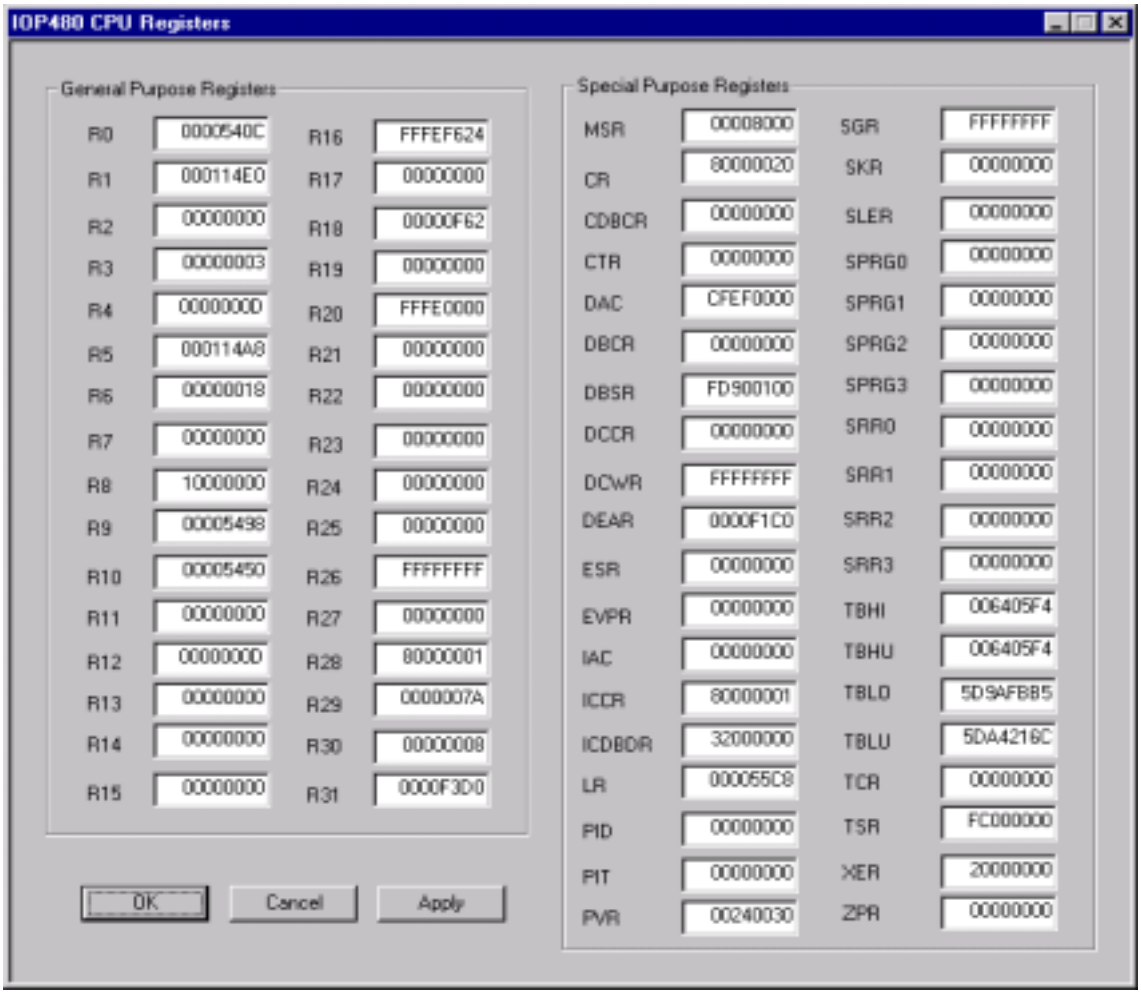


Figure 4-8. IOP 480 CPU Registers Group

© PLX Technology, Inc. All rights reserved.

4.1.8 IOP 480 EEPROM Values

The following figure shows the default values of the EEPROM on the IOP 480 RDK board. These values may be used as the default values. This dialog box allows one to save the EEPROM values to a file and to load values from a file. Clicking on the <OK> and <Apply> button will write the current values to your EEPROM.

Note: You should be very careful before writing any value to the EEPROM. Wrong EEPROM values can cause your board to crash and it may not reboot. You should save the default values to a file on a floppy diskette as a backup.

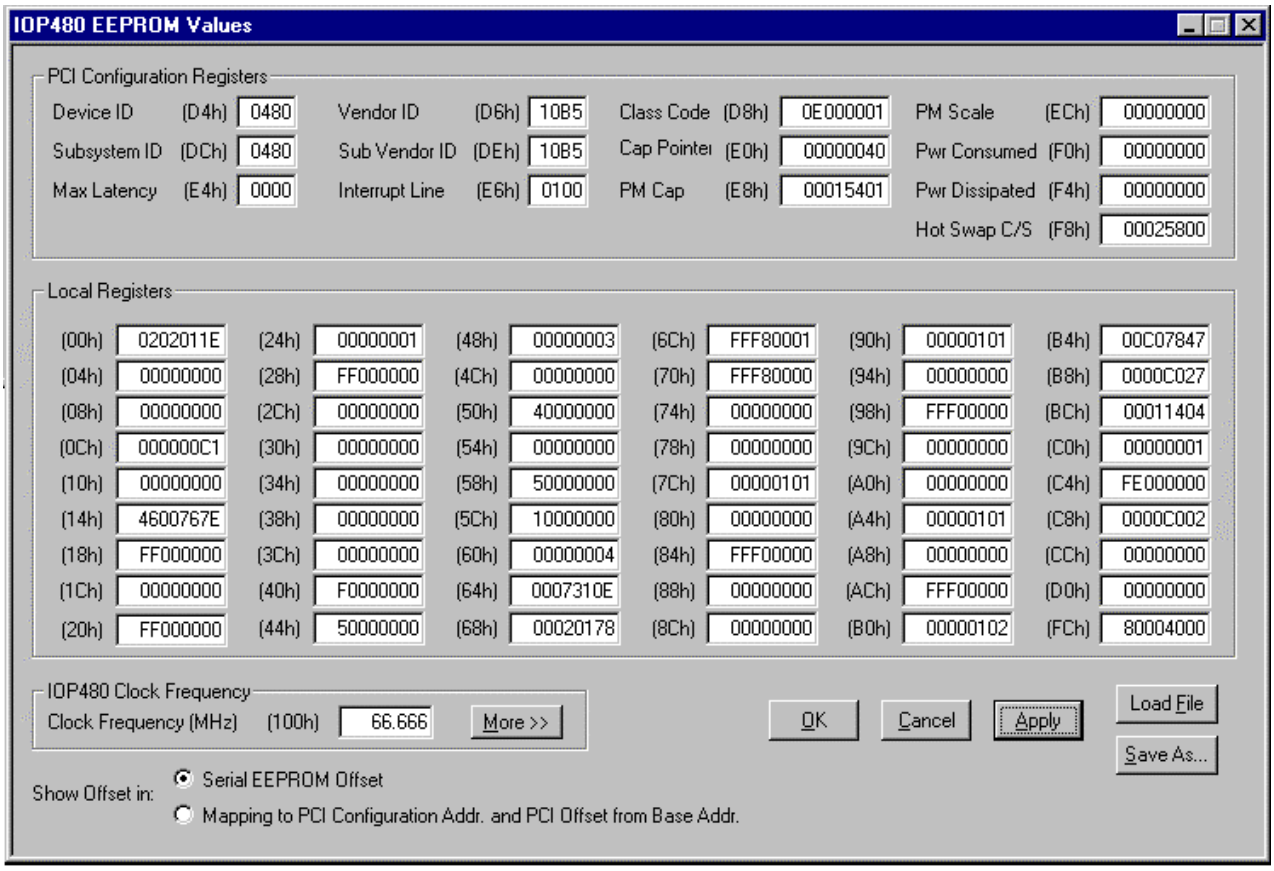
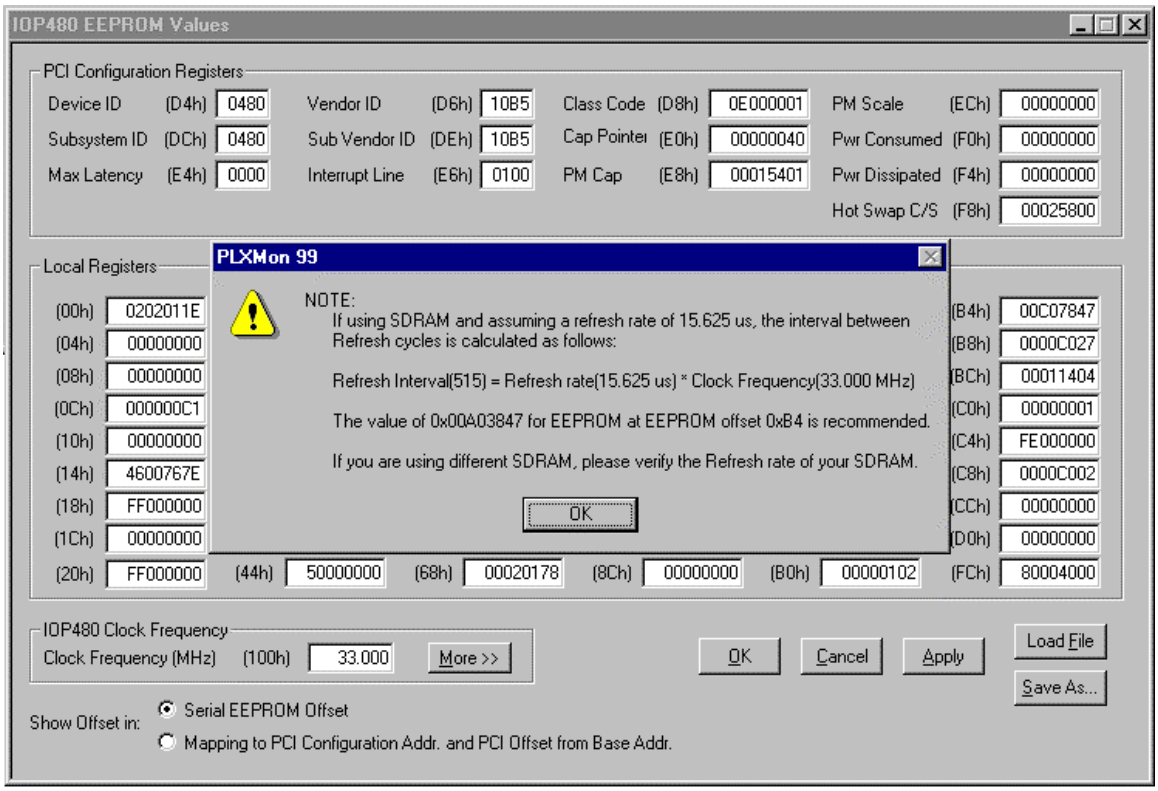


Figure 4-9. IOP 480 EEPROM values

© PLX Technology, Inc. All rights reserved.

By default, the IOP 480RDK board is configured for 66.66MHz frequency. If your board is designed to work at a different frequency, you can change the frequency in the PLXMon 2000 EEPROM screen and you will get a pop up message window as shown below:



Note: It is recommended that you provide the correct value for the type of SDRAM you are using on your board.

© PLX Technology, Inc. All rights reserved.

5. The PCI 9054 Register Set

Each of the PCI 9054's register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers when required.

5.1 The Register Group Dialog Boxes

The PLXMon 2000 toolbar contains five buttons for register accesses. They are for PCI Configuration Registers (PCR), Local Configuration Registers (LCR), RunTime Registers (RTR), DMA Registers (DMA), and Messaging Queue Registers (MQR).

5.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in the PCI Configuration Registers dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and check boxes, also in Figure 5-1, indicate the current settings of the register bit fields. To update the contents of the dialog box push the Refresh button.

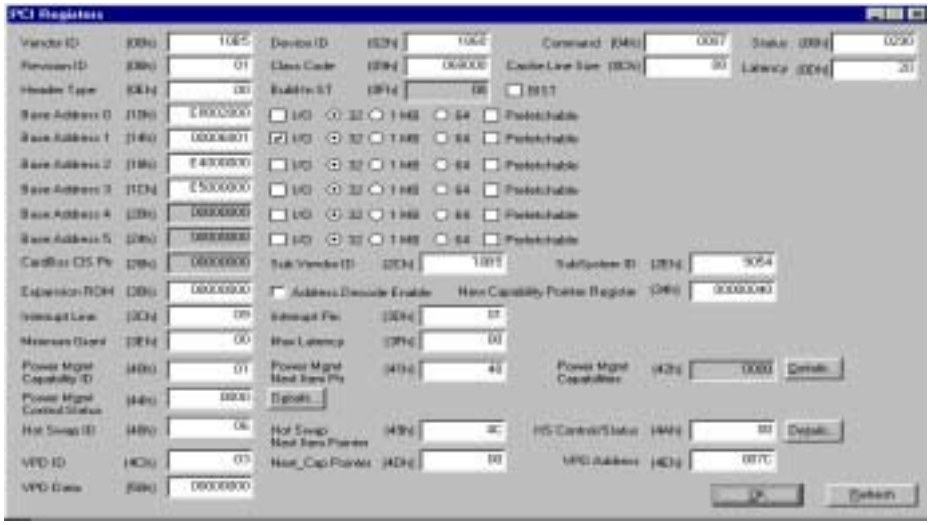


Figure 5-1. PCI Configuration Registers Dialog Box for PCI 9054

© PLX Technology, Inc. All rights reserved.

Power Management Capabilities

This dialog box seen in Figure 5-2 displays Power Management setup attributes that are read-only, or that can only be modified from the IOP side. If this dialog box is opened in Serial Mode, then certain values can be changed.

Figure 5-2. Power Management Capabilities Dialog Box

Power Management Control/Status Register

In Figure 5-3 are the bits that handle the operation of Power Management on the PCI 9054.

Figure 5-3. Power Management CSR Dialog Box

Hot Swap Control/Status Register

The Control and Status bits for Hot Swapping are found here in Figure 5-4. All of these values can only be written from the PCI side.

Figure 5-4. Hot Swap CSR Dialog Box

© PLX Technology, Inc. All rights reserved.

5.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes as seen in Figure 5-5. The size text box reflects the value (in bytes) of the associated register. The memory size is calculated from corresponding register values and cannot be modified directly. To change the memory size, modify the associated register.

Seven registers within the Local Configuration Register Group have a more detailed dialog box and are as follows:

- The Mode/DMA Arbitration dialog box
- The Endian Descriptor dialog box
- The Miscellaneous Control Register dialog box
- The Region 0/Exp ROM dialog box
- The DM PCI Remap dialog box
- The DM Config IO Address dialog box
- The Region 1 Descriptor dialog box

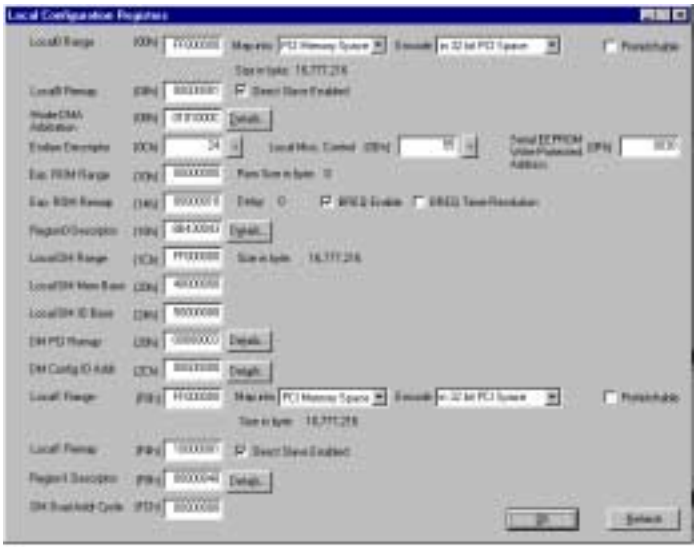


Figure 5-5. Local Configuration Registers Dialog Box

The Mode/Arbitration Dialog Box

The Mode/Arbitration dialog box provides information on the current value of the Local/DMA Arbitration register and allows modification of that value (see Figure 5-6).

Mode / Arbitration Registers
Local Bus Latency Timer Enable
Local Bus Pause Timer Enable
Local Bus BREQ Enable
Local Bus Direct Slave Give up Bus Mode
Direct Slave LLOCK.o# Enable
PCI Request Mode
PCI Rev 2.1 Mode
PCI Read No Write Mode
PCI Read With Write Flush Mode
Gate the Local Bus Latency Timer with BR
PCI Read No Flush Mode
Subsystem ID and Subsystem Vendor ID
Local Bus Latency
0C
Local Bus Pause Timer
00
DMA Channel Priority
Rotational priority scheme
Channel 0 has priority
Channel 1 has priority
BIGEND# / WAIT# Input/Output Select
Signal with wait functionality
Functionality of Big Endian input
OK
Cancel

Figure 5-6. Mode/Arbitration Dialog Box

© PLX Technology, Inc. All rights reserved.

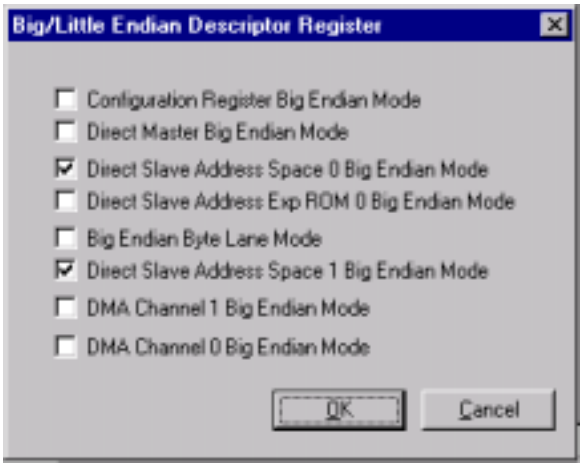

Big/Little Endian Descriptor Register
Configuration Register Big Endian Mode
Direct Master Big Endian Mode
Direct Slave Address Space 0 Big Endian Mode
Direct Slave Address Exp ROM 0 Big Endian Mode
Big Endian Byte Lane Mode
Direct Slave Address Space 1 Big Endian Mode
DMA Channel 1 Big Endian Mode
DMA Channel 0 Big Endian Mode
OK
Cancel

Figure 5-7. Endian Descriptor Dialog Box

Endian Descriptor Dialog Box

The Endian Descriptor dialog box provides information on the current value of the Big/Little Endian Descriptor register and allows modification of that value (see Figure 5-7).

Local Miscellaneous Control Register Dialog Box

This dialog box contains bit controls for the miscellaneous functions of the PCI 9054 (see Figure 5-8).

These functions include:

- Base Address Register 1 support
- Init Done bit signal to BIOS
- Direct Master Enables
- Error interrupt Masks

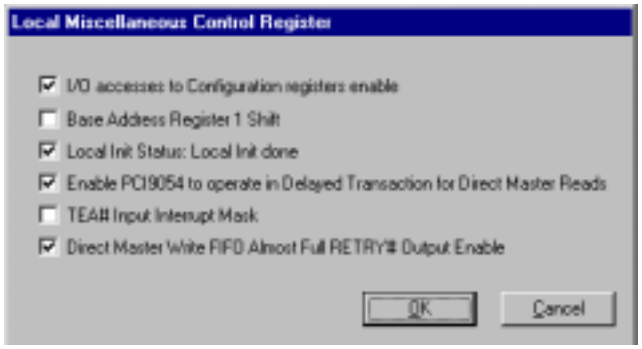

Local Miscellaneous Control Register
I/O accesses to Configuration registers enable
Base Address Register 1 Shift
Local Init Status: Local Init done
Enable PCI9054 to operate in Delayed Transaction for Direct Master Reads
TEA# Input Interrupt Mask
Direct Master Write FIFO Almost Full RETRY# Output Enable
OK
Cancel

Figure 5-8. Local Miscellaneous Control Register Dialog Box

© PLX Technology, Inc. All rights reserved.

The Local Space 0/Exp ROM Dialog Box

The Region 0 descriptor provides information on the current value of the Local Address Space 0/Expansion ROM Bus Region Descriptor register and allows modification of that value

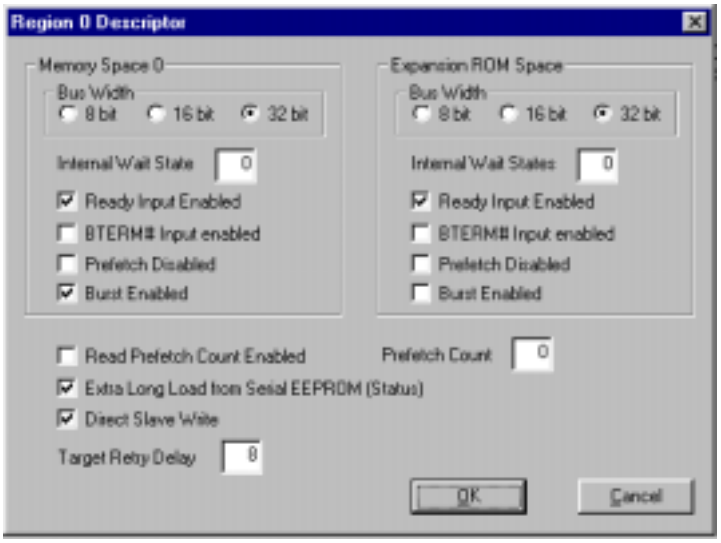


Figure 5-9. Region 0/ROM Descriptor Dialog Box

The DM PCI Remap Dialog Box

The DM PCI Remap dialog box provides information on the current value of the PCI Base Address (Remap) Register for Direct Master to PCI Memory and allows modification of that value (see Figure 5-10).

Figure 5-10. Direct Master Remap Dialog Box

© PLX Technology, Inc. All rights reserved.

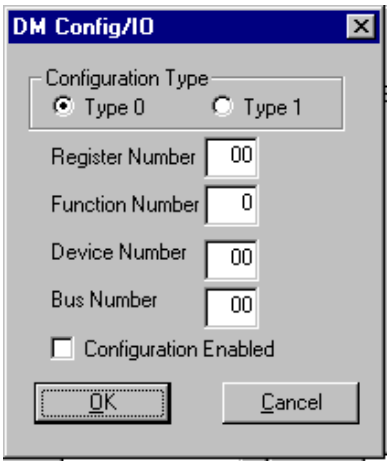


Figure 5-11. Direct Master Config. Dialog box

The DM Configuration I/O Address Dialog Box

The DM Configuration I/O Address dialog box provides information on the current value of the PCI configuration Address Register for Direct Master to PCI I/O-CFG and allows modification of that value (see Figure 5-11).

The Local Space 1 Dialog Box

The Region 1 dialog box provides information on the Local Address Space 1 Bus Region Descriptor register and allows modification of that value (see Figure 5-12).

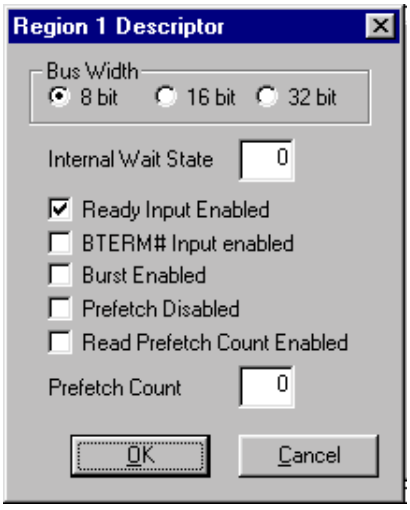


Figure 5-12. Region 1 Descriptor Dialog Box

© PLX Technology, Inc. All rights reserved.

5.1.3 The Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays, and allows modification of, the current register values of the Runtime Registers. See figure below.

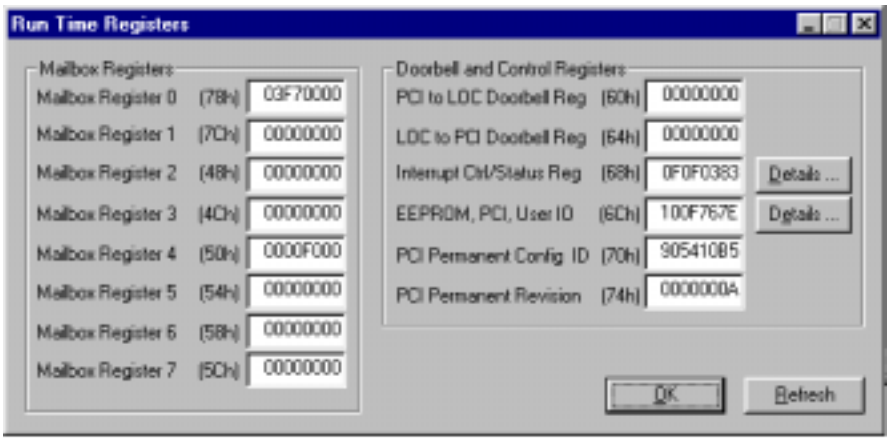

Run Time Registers
Mailbox Registers
Mailbox Register 0 (78h) 03F70000
Mailbox Register 1 (7Ch) 00000000
Mailbox Register 2 (48h) 00000000
Mailbox Register 3 (4Ch) 00000000
Mailbox Register 4 (50h) 0000F000
Mailbox Register 5 (54h) 00000000
Mailbox Register 6 (58h) 00000000
Mailbox Register 7 (5Ch) 00000000
Doorbell and Control Registers
PCI to LOC Doorbell Reg (60h) 00000000
LOC to PCI Doorbell Reg (64h) 00000000
Interrupt Ctrl/Status Reg (68h) 0F0F0383
Details ...
EEPROM, PCI, User IO (6Ch) 100F767E
Details ...
PCI Permanent Config ID (70h) 90541085
PCI Permanent Revision (74h) 0000000A
OK
Refresh

Figure 5-13. Run Time Registers Dialog Box

The Interrupt Control/Status Register Dialog Box

The Interrupt Control/Status Register Dialog Box provides information on the current value of the Interrupt Control/Status register.

The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

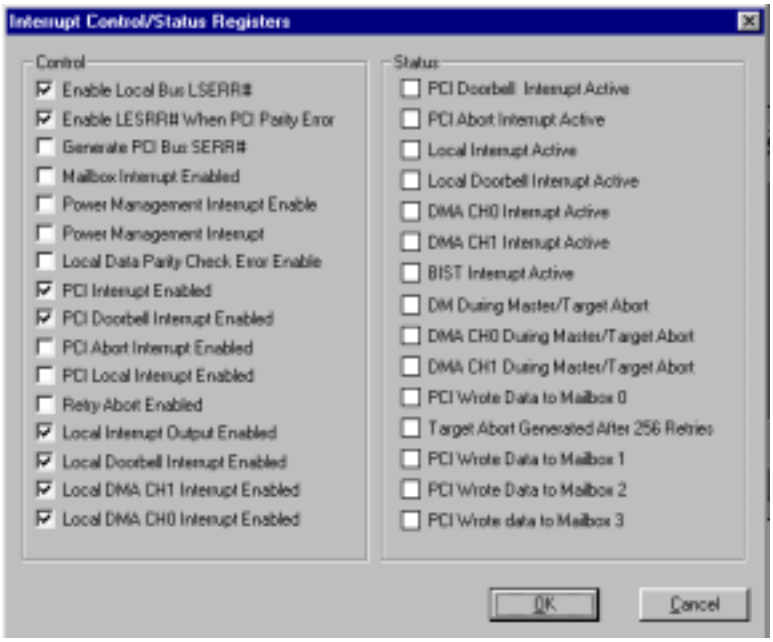

Interrupt Control/Status Registers
Control
Enable Local Bus LSERR#
Enable LESRR# When PCI Parity Error
Generate PCI Bus SERR#
Mailbox Interrupt Enabled
Power Management Interrupt Enable
Power Management Interrupt
Local Data Parity Check Error Enable
PCI Interrupt Enabled
PCI Doorbell Interrupt Enabled
PCI Abort Interrupt Enabled
PCI Local Interrupt Enabled
Retry Abort Enabled
Local Interrupt Output Enabled
Local Doorbell Interrupt Enabled
Local DMA CH1 Interrupt Enabled
Local DMA CH0 Interrupt Enabled
Status
PCI Doorbell Interrupt Active
PCI Abort Interrupt Active
Local Interrupt Active
Local Doorbell Interrupt Active
DMA CH0 Interrupt Active
DMA CH1 Interrupt Active
BIST Interrupt Active
DM During Master/Target Abort
DMA CH0 During Master/Target Abort
DMA CH1 During Master/Target Abort
PCI Wrote Data to Mailbox 0
Target Abort Generated After 256 Retries
PCI Wrote Data to Mailbox 1
PCI Wrote Data to Mailbox 2
PCI Wrote data to Mailbox 3
OK
Cancel

Figure 5-14. Interrupt Control and Status Dialog Box

© PLX Technology, Inc. All rights reserved.

The EEPROM, PCI, User IO Dialog Box

The EEPROM, PCI, User IO dialog box provides information on the current contents of the EEPROM Control, PCI Command Codes, User I/O Control, Init Control Register and allows modification of that value (see Figure 5-15). The Status section contained in this dialog box contains values that cannot be modified.

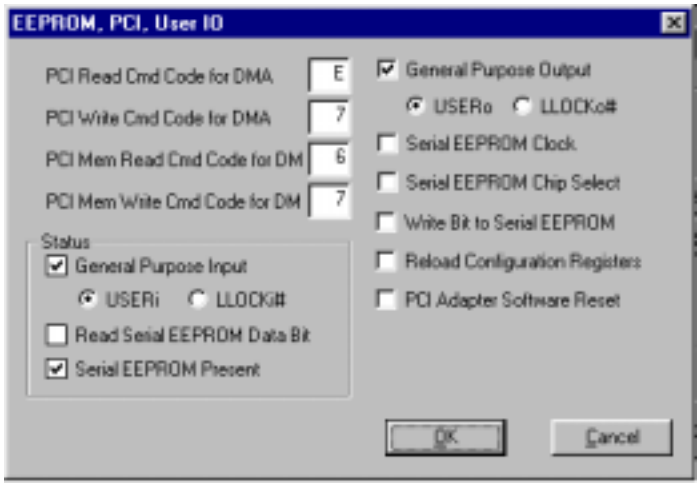


Figure 5-15. EEPROM PCI User IO Dialog Box

5.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for both DMA channels (see Figure 5-16).

The Start and Abort Transfer buttons initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Channel Enable bit enables DMA transfers and activates the Start, Abort, and Clear Interrupt buttons.

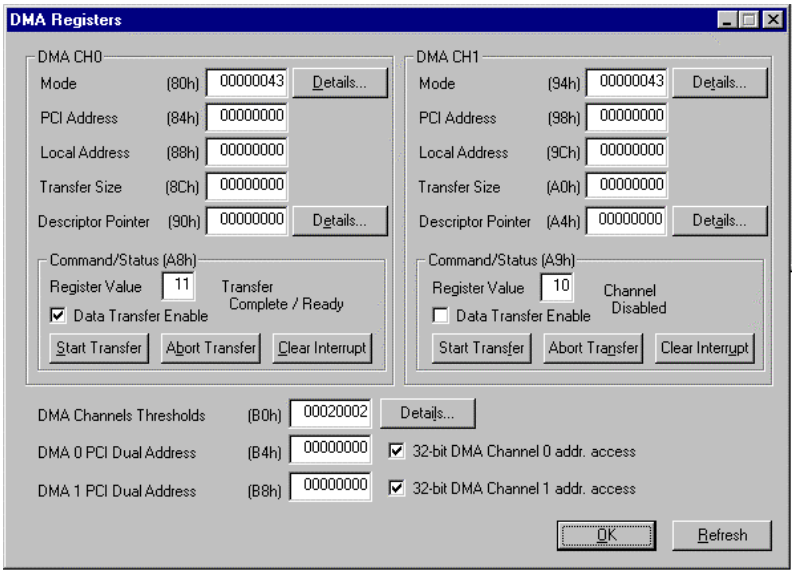
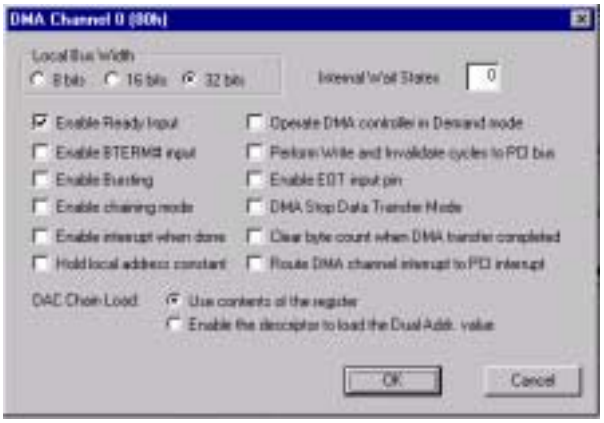


Figure 5-16. Local DMA Registers Dialog Box

© PLX Technology, Inc. All rights reserved.

The DMA Mode Dialog Box



The DMA Mode dialog box provides information on the current value of the DMA Channel's Mode Register and allows modification of that value (see Figure 5-17).

Figure 5-17. DMA Mode Dialog Box

The Descriptor Pointer Dialog Box

The Descriptor Pointer dialog box provides information on the current value of the DMA Channel's Descriptor Pointer Register and allows modification of that value (see Figure 5-18).

Figure 5-18. DMA Descriptor Pointer Dialog Box

The DMA Channels Threshold Dialog Box

The DMA Channels Threshold dialog box provides information on the current value of the DMA Threshold Register and allows modification of that value (see Figure 5-19).

Figure 5-19. DMA Thresholds Dialog Box

© PLX Technology, Inc. All rights reserved.

5.1.5 The Messaging FIFO Register Group Dialog Box

The Messaging FIFO Register Group dialog box contains the current values for the Messaging FIFO Registers as shown in Figure 5-20.

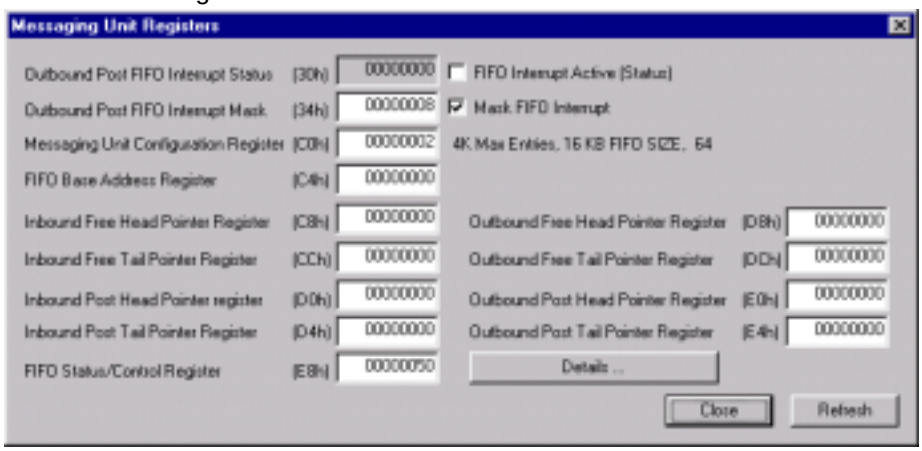

Messaging Unit Registers
Outbound Post FIFO Interrupt Status (30h) 00000000
FIFO Interrupt Active (Status)
Outbound Post FIFO Interrupt Mask (34h) 00000008
Mask FIFO Interrupt
Messaging Unit Configuration Register (C0h) 00000002
4K Max Entries, 16 KB FIFO SIZE, 64
FIFO Base Address Register (C4h) 00000000
Inbound Free Head Pointer Register (C8h) 00000000
Outbound Free Head Pointer Register (D8h) 00000000
Inbound Free Tail Pointer Register (CCh) 00000000
Outbound Free Tail Pointer Register (DCh) 00000000
Inbound Post Head Pointer register (D0h) 00000000
Outbound Post Head Pointer Register (E0h) 00000000
Inbound Post Tail Pointer Register (D4h) 00000000
Outbound Post Tail Pointer Register (E4h) 00000000
FIFO Status/Control Register (E8h) 00000050
Details ...
Close
Refresh

Figure 5-20. Messaging Unit Registers Dialog Box

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified.

The FIFO Status/Control Register Dialog Box

The FIFO Status/Control Register dialog box provides information on the current value of the Queue Status/Control Register and allows modification of that value (see Figure 5-21).

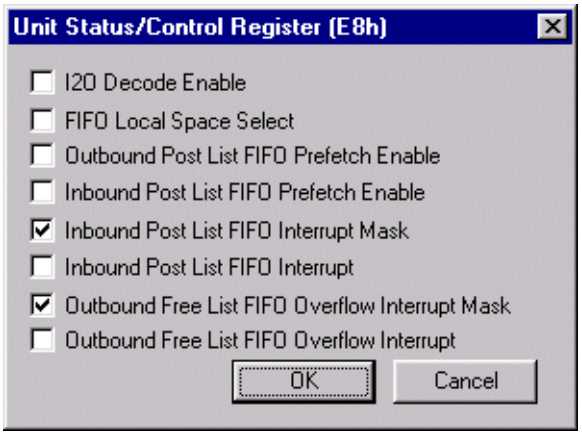

Unit Status/Control Register (E8h)
I2O Decode Enable
FIFO Local Space Select
Outbound Post List FIFO Prefetch Enable
Inbound Post List FIFO Prefetch Enable
Inbound Post List FIFO Interrupt Mask
Inbound Post List FIFO Interrupt
Outbound Free List FIFO Overflow Interrupt Mask
Outbound Free List FIFO Overflow Interrupt
OK
Cancel

Figure 5-21. Status/Control Register Dialog Box

© PLX Technology, Inc. All rights reserved.

6. The PCI 9080 Register Set

Each of the PCI 9080's register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers if necessary.

6.1 The Register Group Dialog Boxes

The PLXMon 2000 toolbar contains five buttons for register accesses. They are for PCI Configuration Registers (PCR), Local Configuration Registers (LCR), RunTime Registers (RTR), DMA Registers (DMA), and Messaging Queue Registers (MQR).

6.1.1 PCI Configuration Register Group Dialog Box

The grayed text, in Figure 6-1, on the `PCI Configuration Registers` dialog box indicates that the values cannot be modified using this dialog box. The radio buttons and check boxes indicate the current settings of the register bit fields. To update the contents of the dialog box push the `Refresh` button.

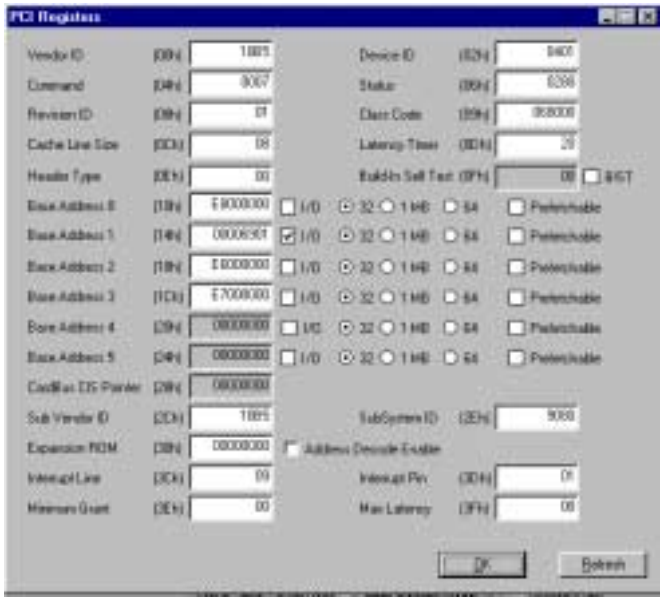


Figure 6-1. PCI Configuration Registers Dialog Box

© PLX Technology, Inc. All rights reserved.

6.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes, in Figure 6-2, respectively. The `size` text box reflects the value (in bytes) of the associated register. The memory size is calculated from the corresponding register value and cannot be modified directly. To change the memory size, modify the associated register.

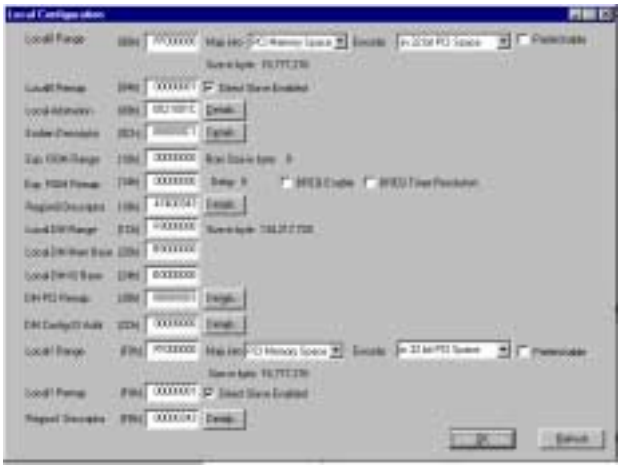


Figure 6-2. Local Configuration Registers Dialog Box

Six registers within the Local Configuration Register Group have a more detailed dialog box and are as follows:

- The Mode/Arbitration dialog box;
- The Endian Descriptor dialog box;
- The Space 0/Exp ROM dialog box;
- The DM PCI Remap dialog box;
- The DM Config I/O Address dialog box; and,
- The Space 1 dialog box.

© PLX Technology, Inc. All rights reserved.

The Mode/Arbitration Dialog Box

The Mode/Arbitration dialog box provides information on the current value Local/DMA Arbitration registers and allows modification of that value (see Figure 6-3).

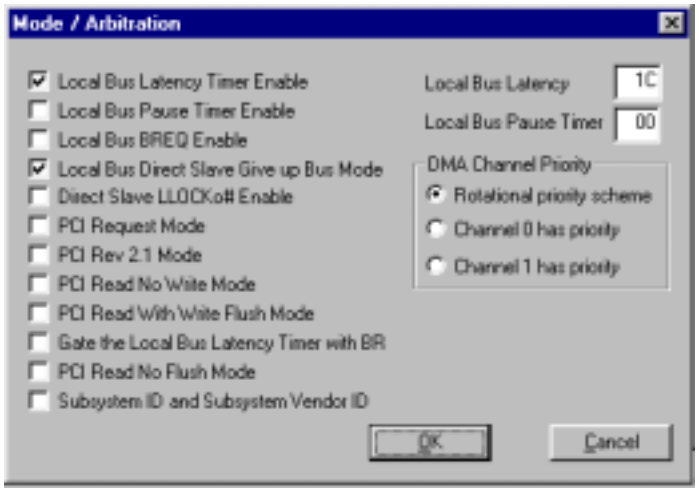

Mode / Arbitration
Local Bus Latency Timer Enable
Local Bus Pause Timer Enable
Local Bus BREQ Enable
Local Bus Direct Slave Give up Bus Mode
Direct Slave LLOCKo# Enable
PCI Request Mode
PCI Rev 2.1 Mode
PCI Read No Write Mode
PCI Read With Write Flush Mode
Gate the Local Bus Latency Timer with BR
PCI Read No Flush Mode
Subsystem ID and Subsystem Vendor ID
Local Bus Latency
1C
Local Bus Pause Timer
00
DMA Channel Priority
Rotational priority scheme
Channel 0 has priority
Channel 1 has priority
OK
Cancel

Figure 6-3. Direct Master Mode / Arbitration Dialog box

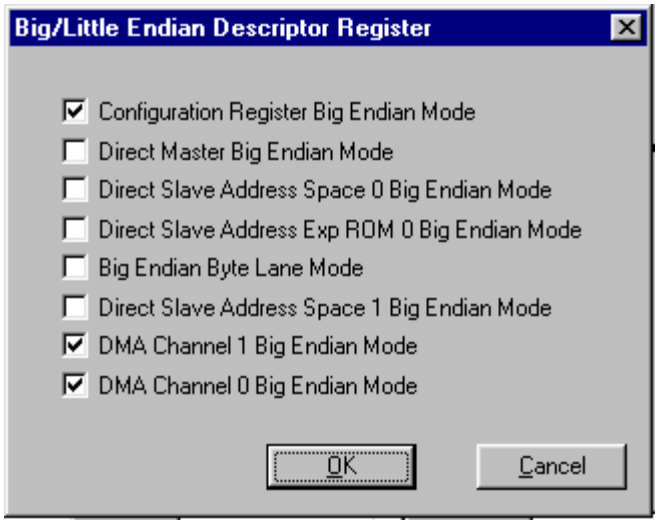

Big/Little Endian Descriptor Register
Configuration Register Big Endian Mode
Direct Master Big Endian Mode
Direct Slave Address Space 0 Big Endian Mode
Direct Slave Address Exp ROM 0 Big Endian Mode
Big Endian Byte Lane Mode
Direct Slave Address Space 1 Big Endian Mode
DMA Channel 1 Big Endian Mode
DMA Channel 0 Big Endian Mode
OK
Cancel

Figure 6-4. Endian Descriptor Dialog Box

Endian Descriptor Dialog Box

The Endian Descriptor dialog box provides information on the current value of the Big/Little Endian Descriptor register and allows modification of that value (see Figure 6-4).

© PLX Technology, Inc. All rights reserved.

The Local Space 0/Exp ROM Dialog Box

The Region 0 Descriptor dialog box provides information on the current value of the Local Address Space 0/Expansion ROM Bus Region Descriptor register and allows modification of that value (see Figure 6-5).

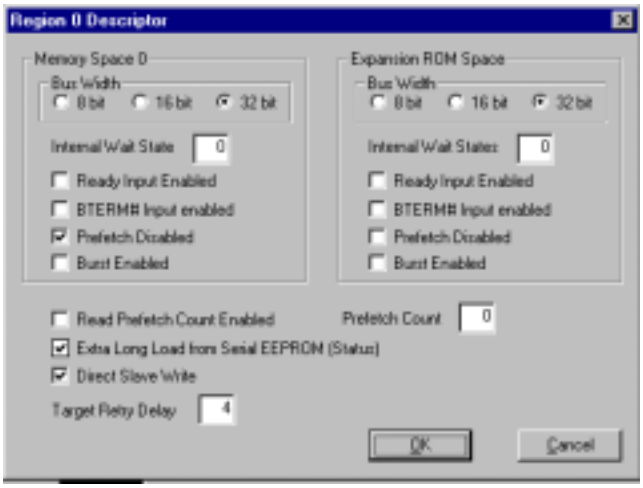


Figure 6-5. Local Space 0/Emp ROM Dialog Box

Figure 6-6. Direct Master PCI Remap Dialog Box

The Direct Master PCI Remap Dialog Box

The Direct Master (DM) PCI Remap dialog box provides information on the current value of the PCI Base Address (Remap) Register for Direct Master to PCI Memory and allows modification of that value (see Figure 6-6).

© PLX Technology, Inc. All rights reserved.

The DM Configuration I/O Address Dialog Box

The DM Configuration I/O Address dialog box provides information on the current value of the PCI configuration Address Register for Direct Master to PCI I/O-CFG and allows for modification of that value (see Figure 6-7).

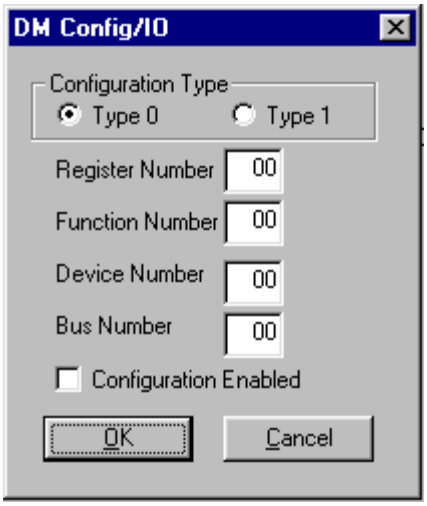


Figure 6-7. Direct Master Configure Dialog Box

The Region 1 Dialog Box

The Region 1 dialog box provides information on the Local Address Space 1 Bus Region Descriptor register and allows modification of that value (see Figure 6-8).

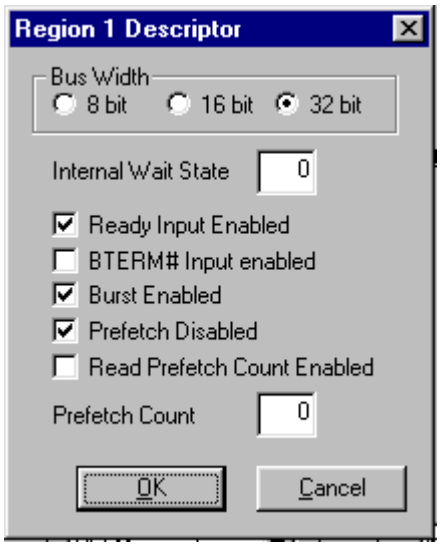
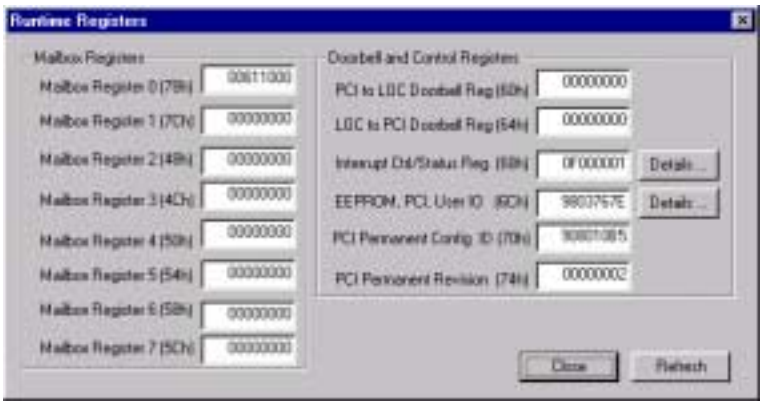


Figure 6-8. Region 1 Descriptor Dialog Box

© PLX Technology, Inc. All rights reserved.

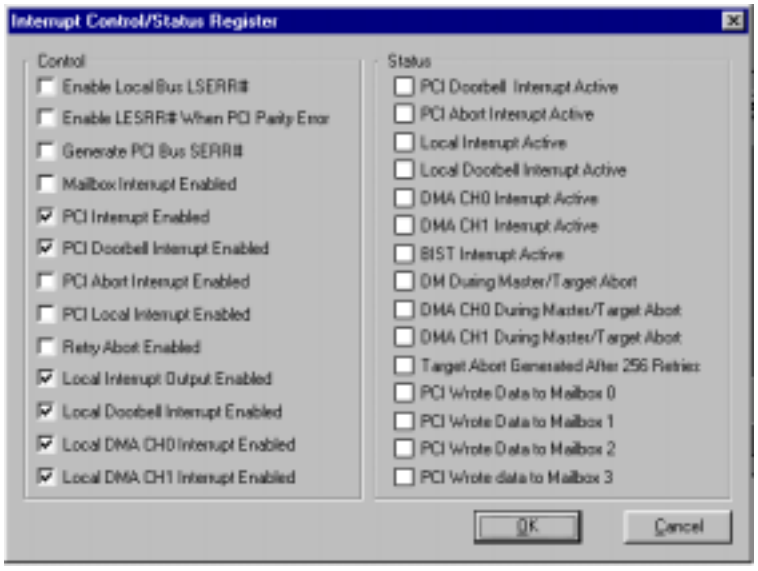
6.1.3 The Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays, and allows modification of, the current register values of the Runtime Registers.



The Interrupt Control/Status Register Dialog Box

The Interrupt Control/Status Register Dialog Box provides information on the current value of the Interrupt Control/Status register.



The information contained in the dialog box is grouped into two categories, the Control bits and the Status bits. The control bits enable triggering of interrupts for certain events, such as DMA events, doorbell events and others. The status bits cannot be modified directly. They show the current status of the various interrupt triggers.

Figure 6-10. Interrupt Control and Status Dialog Box

© PLX Technology, Inc. All rights reserved.

The EEPROM, PCI, User IO Dialog Box

The EEPROM, PCI, User I/O dialog box provides information on the current contents of the EEPROM Control, PCI Command Codes, User I/O Control, Initialization Control Register and allows modification of that value (see Figure 6-11). The Status section contained in this dialog box contains values that cannot be modified.

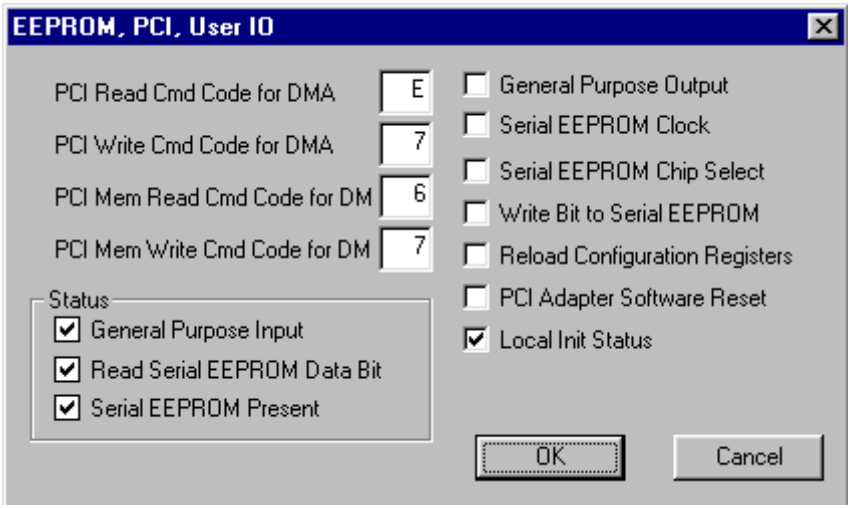


Figure 6-11. EEPROM, PCI, User IO Details Dialog Box

6.1.4 The DMA Register Group Dialog Box

The DMA Register Group dialog box contains the current values for the DMA registers for both DMA channels (see Figure 6-12).

The `Start` and `Abort Transfer` buttons, in Figure 6-12, initiate and terminate the DMA transfer using the current information provided in the DMA registers for the given DMA channel. The Channel `Enable` bit enables DMA transfers and activates the `Start`, `Abort`, and `Clear Interrupt` buttons.

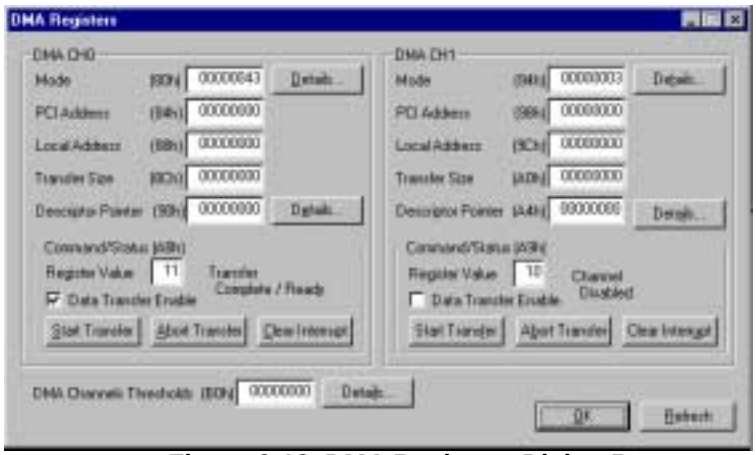


Figure 6-12. DMA Registers Dialog Box

© PLX Technology, Inc. All rights reserved.

The DMA Mode Dialog Box

The DMA Mode dialog box provides information on the current value of the DMA Channel's Mode Register and allows modification of that value (see Figure 6-13).

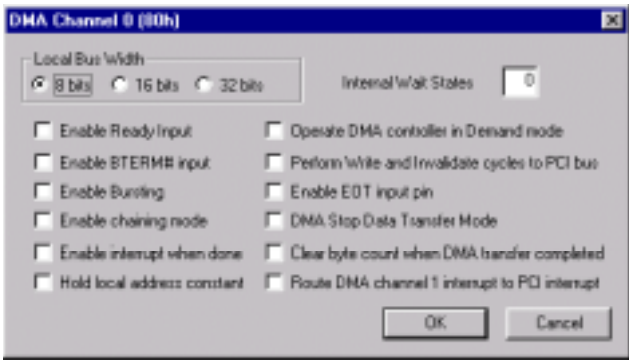
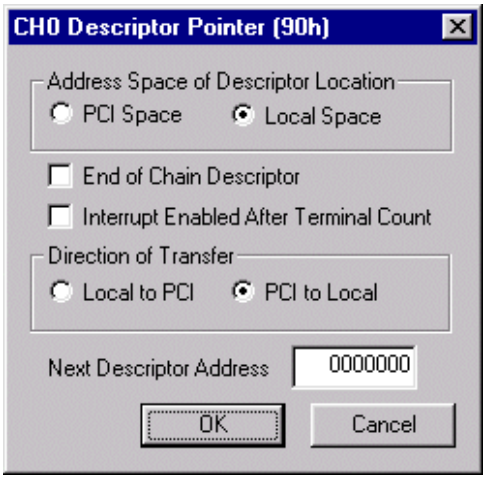


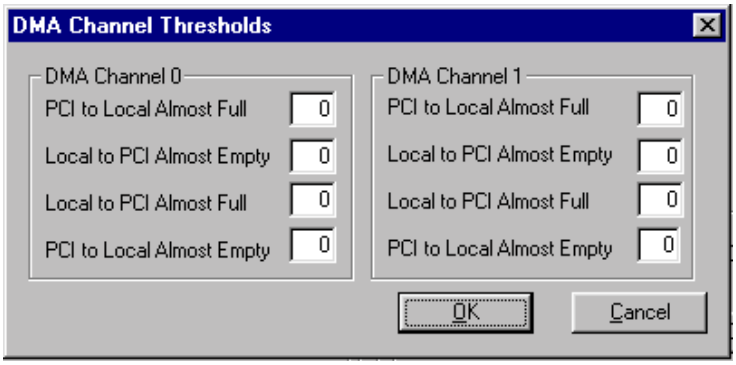
Figure 6-13. DMA Mode Dialog Box



The Descriptor Pointer Dialog Box

The Descriptor Pointer dialog box provides information on the current value of the DMA Channel's Descriptor Pointer Register and allows modification of that value (see Figure 6-14).

Figure 6-14. Descriptor Pointer Dialog Box



The DMA Channels Threshold Dialog Box

The DMA Channels Threshold dialog box provides information on the current value of the DMA Threshold Register and allows modification of that value (see Figure 6-15).

Figure 6-15. DMA Channels Threshold Dialog Box

© PLX Technology, Inc. All rights reserved.

6.1.5 The Messaging FIFO Register Group Dialog Box

The Messaging FIFO Register Group dialog box displays the current values for the Messaging FIFO Registers as shown in Figure 6-16.

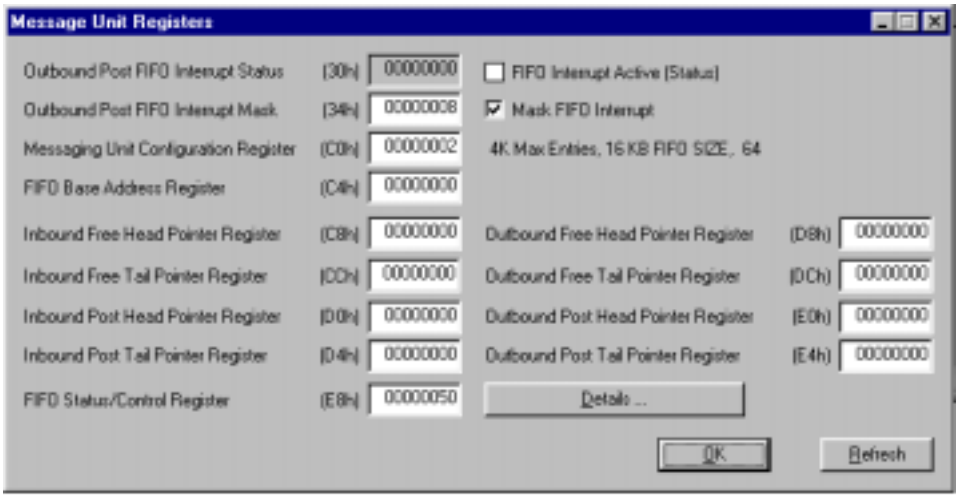

Message Unit Registers
Outbound Post FIFO Interrupt Status (30h) 00000000
FIFO Interrupt Active (Status)
Outbound Post FIFO Interrupt Mask (34h) 00000008
Mask FIFO Interrupt
Messaging Unit Configuration Register (C0h) 00000002
4K Max Entries, 16 KB FIFO SIZE, 64
FIFO Base Address Register (C4h) 00000000
Inbound Free Head Pointer Register (C8h) 00000000
Outbound Free Head Pointer Register (D8h) 00000000
Inbound Free Tail Pointer Register (CCh) 00000000
Outbound Free Tail Pointer Register (DCh) 00000000
Inbound Post Head Pointer Register (D0h) 00000000
Outbound Post Head Pointer Register (E0h) 00000000
Inbound Post Tail Pointer Register (D4h) 00000000
Outbound Post Tail Pointer Register (E4h) 00000000
FIFO Status/Control Register (E8h) 00000050
Details ...
OK
Refresh

Figure 6-16. Messaging Unit Registers Dialog box

All the register values can be modified with the exception of the Outbound Post FIFO Interrupt Status register. This register provides only the status of the Outbound Post FIFO interrupt and cannot be modified.

The FIFO Status/Control Register Dialog Box

The FIFO Status/Control Register dialog box provides information on the current value of the Queue Status/Control Register and allows modification of that value (see Figure 6-17).

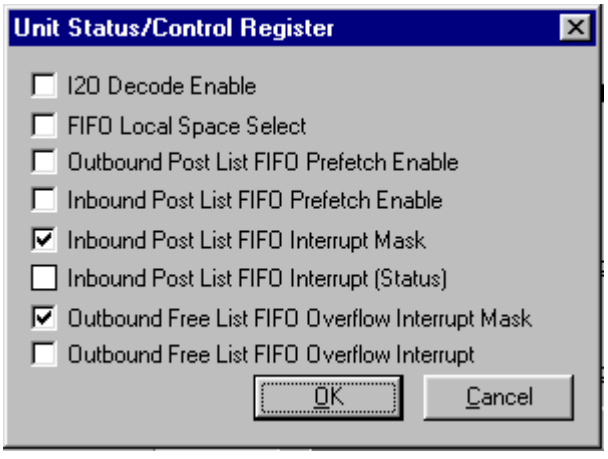

Unit Status/Control Register
I2O Decode Enable
FIFO Local Space Select
Outbound Post List FIFO Prefetch Enable
Inbound Post List FIFO Prefetch Enable
Inbound Post List FIFO Interrupt Mask
Inbound Post List FIFO Interrupt (Status)
Outbound Free List FIFO Overflow Interrupt Mask
Outbound Free List FIFO Overflow Interrupt
OK
Cancel

Figure 6-17. FIFO Status/Control Register Dialog Box

© PLX Technology, Inc. All rights reserved.

7. The PCI 9030 Register Set

Each of the PCI 9030's register groups has a distinct dialog box. Each dialog box has the register values, the register's PCI base addresses, and a description of the register. Some registers have check boxes and radio buttons to help in describing and setting the register values. Additional dialog boxes are available for more complex registers if necessary.

7.1 The Register Group Dialog Boxes

The PLXMon 2000 toolbar contains four buttons for register accesses. They are for PCI Configuration Registers (PCR), Local Configuration Registers (LCR), Chip Select Registers (CSR) and RunTime Registers (RTR).

7.1.1 PCI Configuration Register Group Dialog Box

The PCI Configuration Register Group dialog box contains the current values for the PCI registers. The grayed fields, in Figure 7-1, on the `PCI Configuration Registers` dialog box indicate that the values cannot be modified using this dialog box. The radio buttons and check boxes indicate the current settings of the register bit fields. To update the contents of the dialog box push the `Refresh` button.

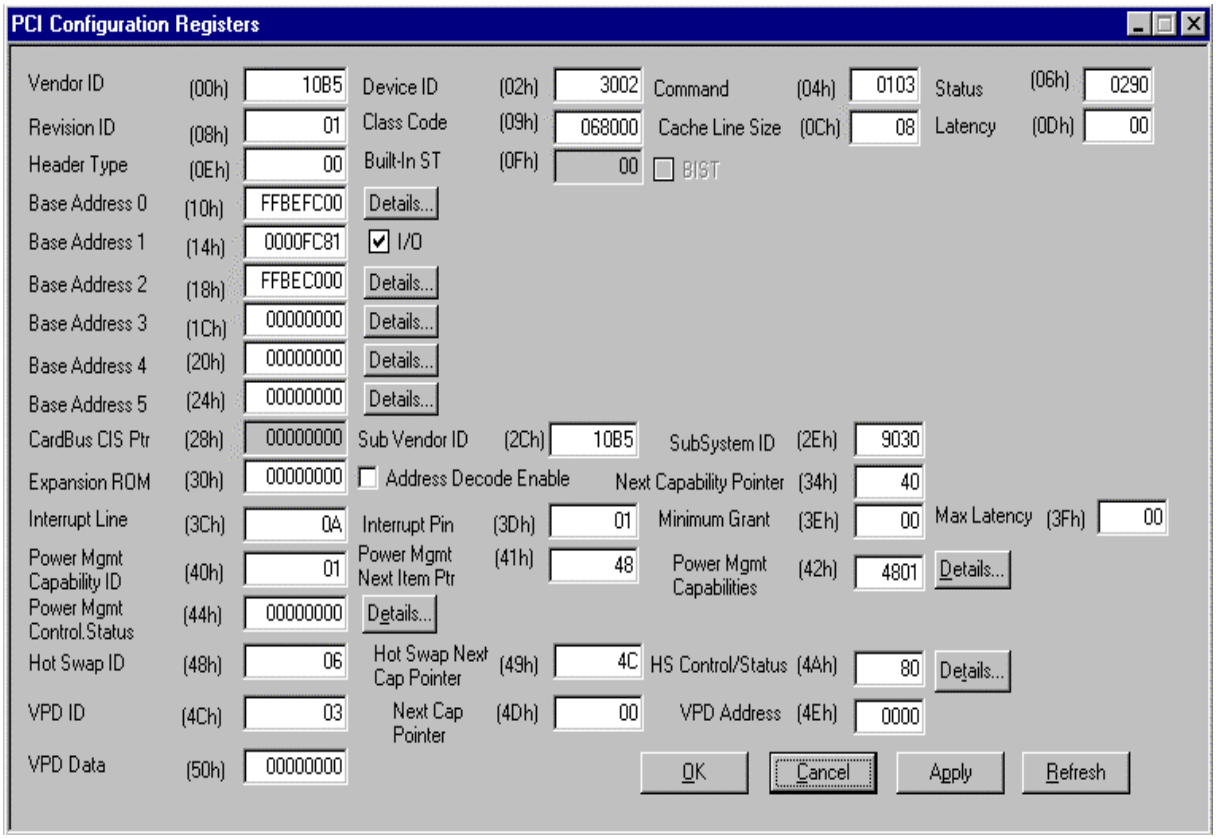


Figure 7-1. PCI Configuration Registers Dialog Box

© PLX Technology, Inc. All rights reserved.

7.1.2 Local Configuration Register Group Dialog Box

The Local Configuration register values are updated through the related edit and dialog boxes, in Figure 7-2, respectively.

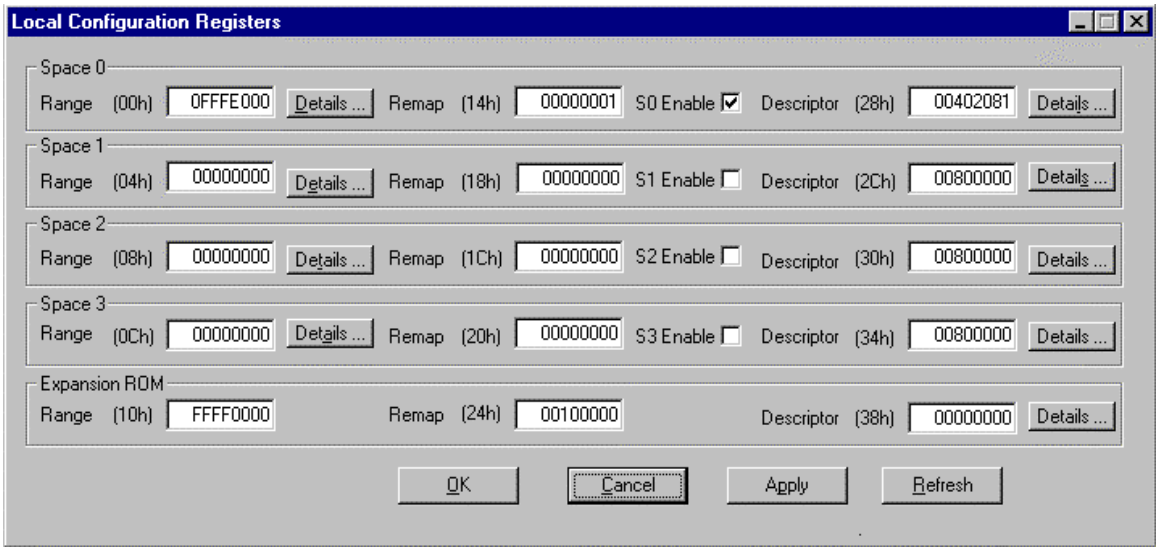

Local Configuration Registers
Space 0
Range (00h) 0FFFE000 Details ... Remap (14h) 00000001 S0 Enable Descriptor (28h) 00402081 Details ...
Space 1
Range (04h) 00000000 Details ... Remap (18h) 00000000 S1 Enable Descriptor (2Ch) 00800000 Details ...
Space 2
Range (08h) 00000000 Details ... Remap (1Ch) 00000000 S2 Enable Descriptor (30h) 00800000 Details ...
Space 3
Range (0Ch) 00000000 Details ... Remap (20h) 00000000 S3 Enable Descriptor (34h) 00800000 Details ...
Expansion ROM
Range (10h) FFFF0000 Remap (24h) 00100000 Descriptor (38h) 00000000 Details ...
OK
Cancel
Apply
Refresh

Figure 7-2. Local Configuration Registers Dialog Box

Detailed dialog boxes for local memory descriptors provide information on the current values of the registers and allow modification of the values (see Figure 7-3).

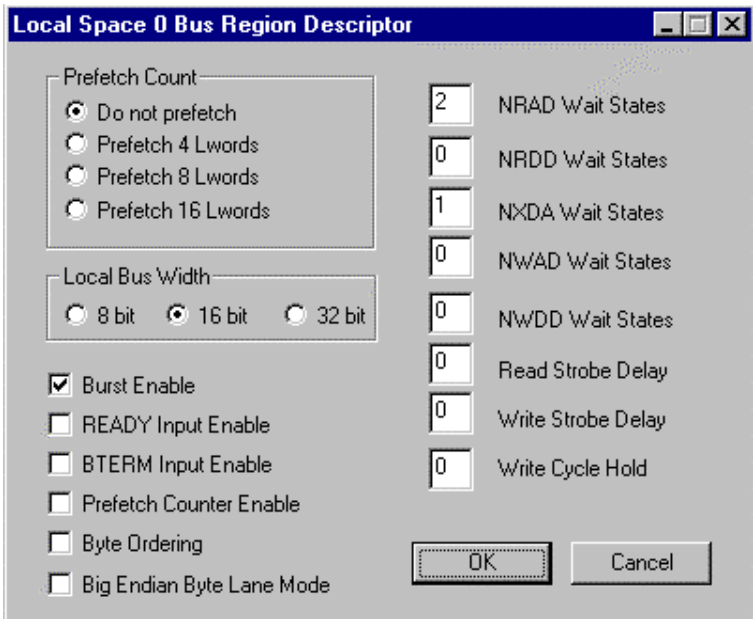


Figure 7-3. Local Memory Bus Region Descriptor Registers Dialog Box

All the register values can be modified with the exception of the reserved bits or Serial EEPROM write only bits (Refer to the PCI 9030 Data Book Chapter 10).

© PLX Technology, Inc. All rights reserved.

7.1.3 Chip Select Register Group Dialog Box

The Chip Select Register Group Dialog Box contains the current values for the Chip Select Registers as shown in Figure 7-4.

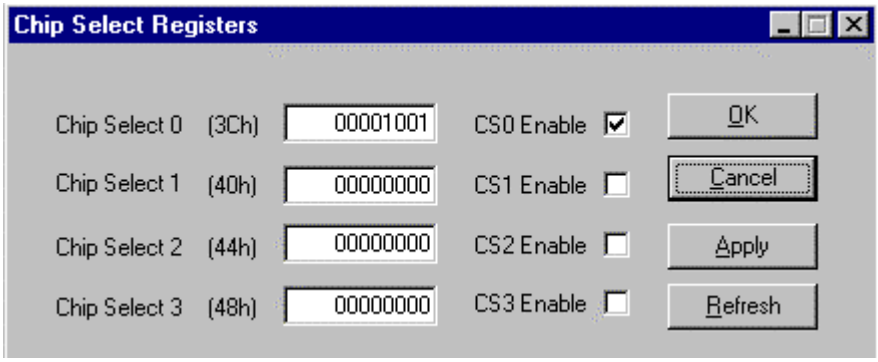


Figure 7-4. Chip Select Registers Dialog Box

All the register values can be modified with the exception that bits 28 to 31 in each chip select register are reserved (Refer to the PCI 9030 Data Book Chapter 10).

7.1.4 Runtime Register Group Dialog Box

The Runtime Register Group dialog box displays, and allows modification of, the current register values of the Runtime Registers. See figure below.

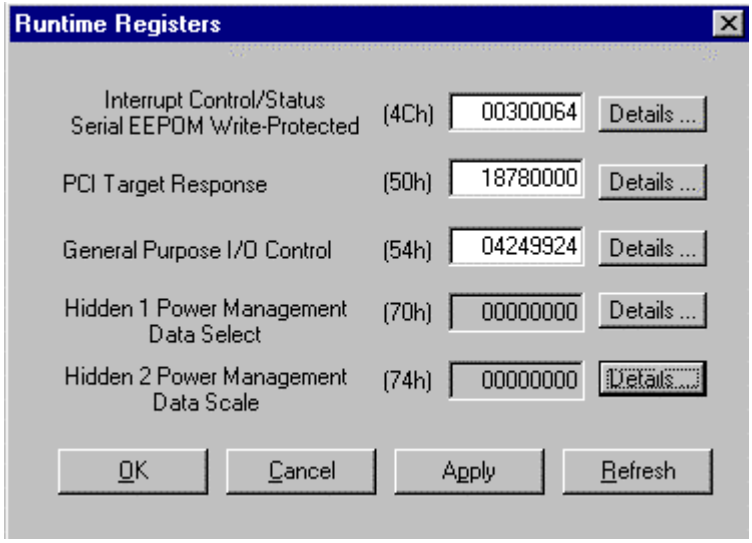

Runtime Registers
Interrupt Control/Status Serial EEPOM Write-Protected (4Ch) 00300064 Details ...
PCI Target Response (50h) 18780000 Details ...
General Purpose I/O Control (54h) 04249924 Details ...
Hidden 1 Power Management Data Select (70h) 00000000 Details ...
Hidden 2 Power Management Data Scale (74h) 00000000 Details ...
OK
Cancel
Apply
Refresh

Figure 7-5. Runtime Registers Dialog Box

Hidden register values are read only. Other runtime register values can be modified with the exception of some read only bits (Refer to the PCI 9030 Data Book Chapter 10).

© PLX Technology, Inc. All rights reserved.

The Interrupt Enable/Status Dialog Box

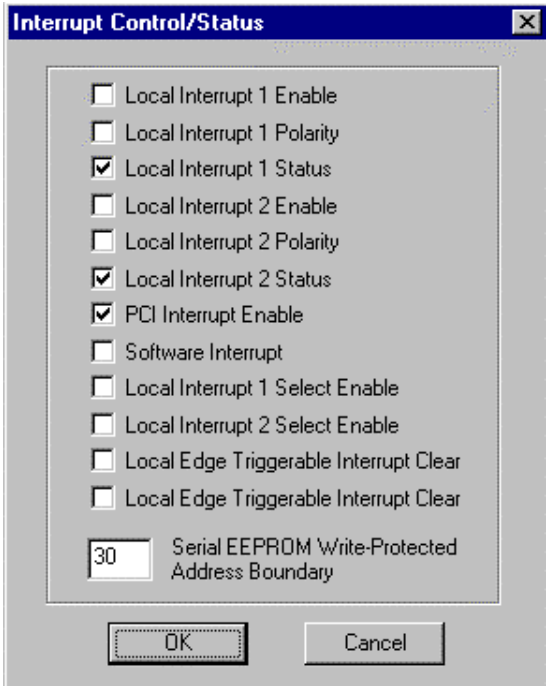
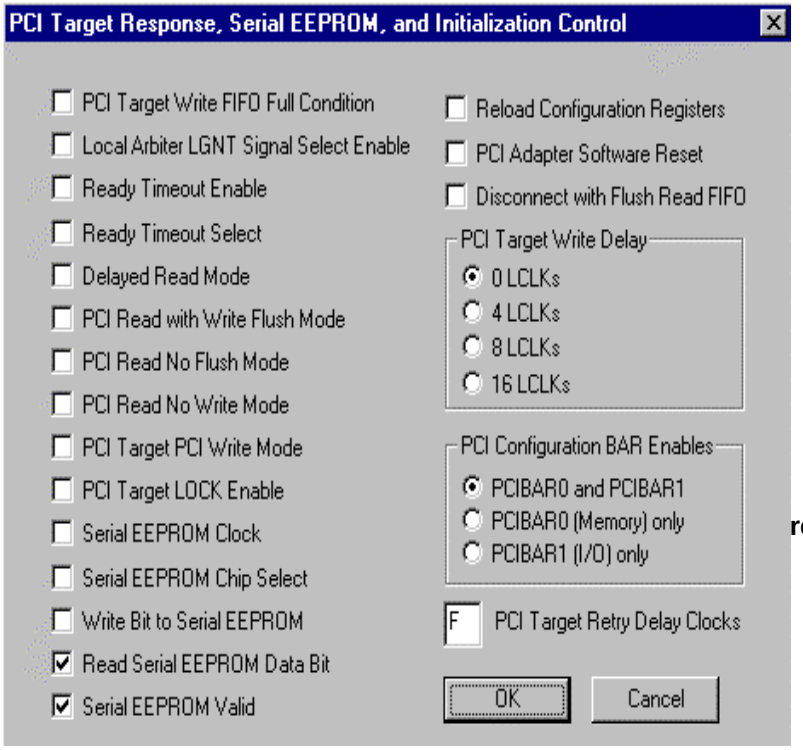


Figure 7-6. Interrupt Enable/Status Dialog Box

The Interrupt Control/Status Dialog Box provides information on the current value of the Interrupt Control/Status register. The enable bits enable triggering of interrupts for certain events. The status bits cannot be modified directly. They show the current status of the various interrupt triggers. To modify a value, click the OK button and then click the Apply or Refresh button on the Runtime Register Dialog Box.

© PLX Technology, Inc. All rights reserved.

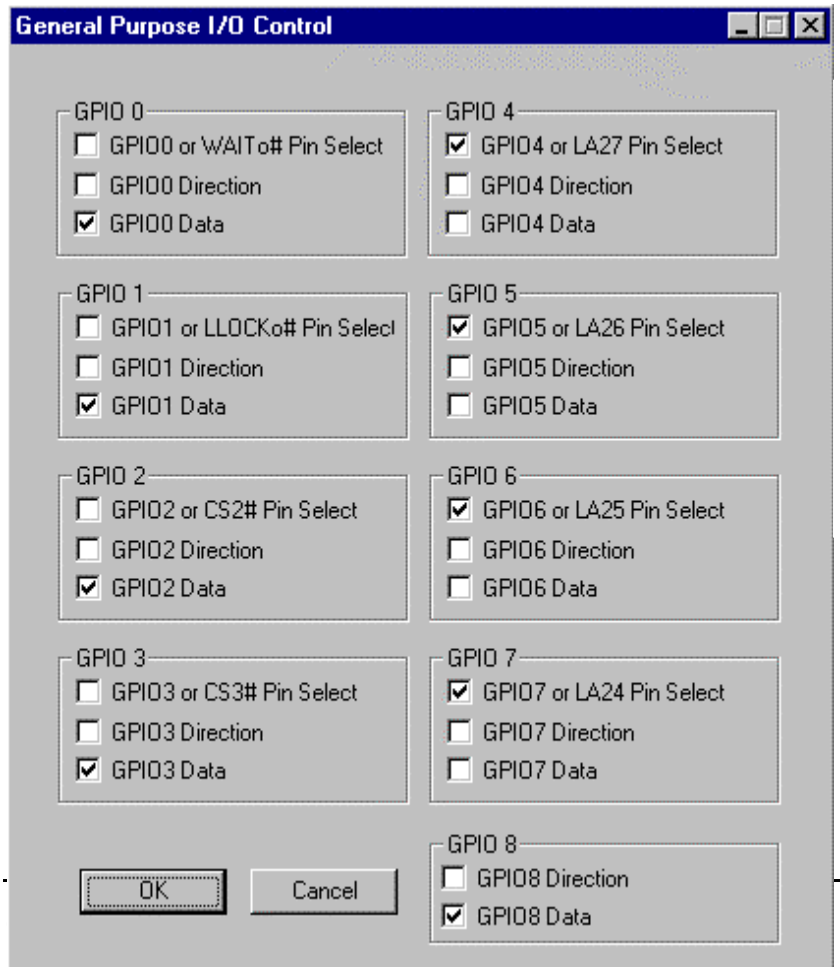
The PCI Target Response Dialog Box



The PCI Target Response Dialog Box provides information on the current value of the PCI Target Response, Serial EEPROM control and Initialization control. To modify a value, click the OK button and then click Apply or Refresh button on Runtime Register Dialog Box.

re 7-7. PCI Target Response Dialog Box

The General Purpose I/O Control Dialog Box



The General Purpose I/O Control Dialog Box displays the current setting of the GPIOs.

To modify a value, click the OK button and then click Apply or Refresh button on Runtime Register Dialog Box.

Figure 7-8. General Purpose I/O Control Dialog Box

© PLX Technology, Inc. All rights reserved.

Appendix A. Troubleshooting

In this section you can find solutions to common problems encountered while using PLXMon 2000. If you encountered a problem that is not listed here, please contact PLX customer support (Section 1.4)

I know I have an RDK in my computer, yet when I start PLXMon 2000, the program will only give me serial access. (WinNT only)

This means the driver was unable to find a "supported" device on your computer. When this happens, the driver will unload itself. Use the event viewer to verify this occurred and to check the cause. Use the driver wizard to add the vendor and device ID of your PCI device to the supported list. Then either restart the computer or manually restart the driver. Instructions for adding a supported device can be found in the SDK User's Manual.

After installing my custom board (the Vendor and Device IDs are my own) the Add New Hardware Wizard in Windows98 or Windows 2000 cannot find my board.

-or-

When adding two different RDK boards at the same time the Add New Hardware Wizard cannot differentiate between them. How do I know which board is which?

The "Add New Hardware Wizard" in Win98 relies on the Vendor and Device IDs of the PCI cards you are inserting. If a custom board is inserted, you must tell the Wizard that the .inf (installation script file) is located in the Inf directory under the Windows system directory. Then select "Unknown PCI XXXX board" depending on the PLX chip that is present on the RDK. The Inf directory is hidden, so make sure you "View all types" within the viewing options of Explorer to find it. Be sure to add new RDKs one at a time to avoid confusing the Wizard.

© PLX Technology, Inc. All rights reserved.

Appendix B. Glossary Of Terms

Back End Monitor (BEM) or (BEM L1)

The Back End Monitor is an embedded program that can be compiled into the embedded software running on a PLX RDK board. Its purpose is to scan the serial input and redirect any data that it determines to be a BEM command. The BEM commands allow reads, writes, and resets of a PLX RDK. For more information about BEM, see the PLX SDK Pro User's Manual.

COFF File Format

Coff files normally are the final data format for a RAM application compiled for use on the PCI and CompactPCI 9054RDK-860RDK boards. The data contained within this file is Big Endian.

IBM-401B Image File

IOP RAM programs compiled for the PCI 9080RDK-401B for RAM will be created in this format. The data is stored in Big-Endian format.

IOP (Input / Output Processor)

This term is interchangeable with the Embedded platform or Local side. This may mean all the software and/or hardware that is on a PLX RDK.

Motorola SRecord

This file format is produced as an intermediate file when compiling code for the PCI and CompactPCI 9054RDK-860. Data is not stored in any particular Endian format.

PCI bus

The PCI bus physically is the location (along with a slot) where the PLX RDK is inserted. The PCI bus can also be given as an address range with data accessible according to the PCI specification.

Host SDK 3.1

This is the current version of PLX's Host Software Development Kit

PLXMon 2000

This application.

© PLX Technology, Inc. All rights reserved.